September/Oktober 2017 | Nr. 557
www.elektormagazine.de

# elektor

LEARN DESIGN SHARE

# Candle2Light

## Kerzenlicht-Effizienzbooster

**In dieser Ausgabe:**
6 Labor-Projekte
7 Homelab-Projekte
3x Grundlagen
und vieles mehr!



**GoNotify**
Ein flexibles
IoT-Sensor-Interface
**Einfach in die Cloud**



**Klein, aber oho:**
**Raspberry Pi Zero W**
Erste Schritte
mit der Mini-Himbeere

**Fernbedienung mit**
**Bluetooth Low Energy**
Mit nur einem Knopfdruck...

**Impedanz-Analyser** • HF-Frontend für FPGA/DSP-Board • **iPhone/iPad-Anwendungen einfach entwickeln** • Muskelsteuerung für Servo-Motoren • **Erste Schritte mit LoRa** • LED-Treiber für 12 V • **ESP32 für Power-User** • Grundlagen: Digitale Korrelation • **Objekte auf Touchscreens erfassen** • **Elektor-Video-Olympiade** • Jammer-Alarm • **Kinetische Skulptur** • Marderscheuche • **Der I²C-Bus: Bausteine und Fehleranalyse** • Retronik: Als die Bilder sprechen lernten • **Q & A: Qualität von Audio-Steckern** • Bemerkenswerte Bauteile: Bandgap-Referenz

G3078 E

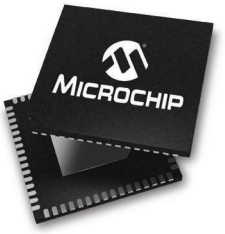
(D) € 14,90 | CHF 27,00 | (A, B, L) € 16,00

4 198630 314904 10

# Mikrocontroller schließt die Grafik-Lücke

## Erste MCU mit 2D-Grafikverarbeitungseinheit und DDR2-Speicher

Die branchenweit erste MCU, die eine 2D Graphics Processing Unit (GPU) und integrierten DDR2-Speicher vereint, sorgt für beste Grafik mit höherer Farbauflösung bei größeren Displays.




Der dreischichtige Grafikcontroller der 32-Bit PIC32MZ-DA-Familie steuert 24-Bit Super Extended Graphics Array (SXGA) Farbdisplays bis 12 Zoll an, während umfangreicher Speicher bis 32 MB On-Chip-DRAM oder 128 MB extern adressierbares DRAM zur Verfügung stehen.

Die PIC32MZ DA MCUs überbrücken die Grafikleistungslücke, um komplexe Grafiken mit den einfach zu bedienenden Entwicklungstools MPLAB® X IDE und MPLAB Harmony sowie Software von Microchip zu erstellen.

microchip DIRECT
www.microchipdirect.com

MICROCHIP

**www.microchip.com/PIC32MZDA**

Der Name Microchip, das Microchip-Logo und MPLAB sind eingetragene Warenzeichen der Microchip Technology Incorporated in den USA und in anderen Ländern. REAL ICE ist eine Marke der Microchip Technology Inc. in den USA und in anderen Ländern. Alle anderen hier erwähnten Marken sind im Besitz der jeweiligen Eigentümer. © 2017 Microchip Technology Inc. Alle Rechte vorbehalten. DS60001490A. MEC2158Ger07/17

**Impressum**

48. Jahrgang, Nr. 557
September/Oktober 2017

Erscheinungsweise: 12 x jährlich
*(Elektor Doppelheft im Wechsel mit dem Elektor Business Magazin)*

**Verlag**
Elektor-Verlag GmbH
Kackertstraße 10
52072 Aachen
Tel. 02 41/955 09 190
Fax 02 41/955 09 013

Technische Fragen bitten wir per E-Mail an redaktion@elektor.de zu richten.

**Hauptsitz des Verlags**
Elektor International Media
Postbus 11, 6114 ZG Susteren
Niederlande

**Anzeigen**:
Margriet Debeij (verantwortlich)
Tel. 02 41/955 09 174 / Fax 02 41/955 09 013
Mobil: +49 (170) 550 539 6
E-Mail: margriet.debeij@eimworld.com

Julia Grotenrath
Tel. 02 41/955 09 177 / Fax 02 41/955 09 013
Mobil: +31 6 510 530 39
E-Mail: julia.grotenrath@eimworld.com

Tanja Pohlen
Tel. 02 41/955 09 186
E-Mail: tanja.pohlen@eimworld.com

*Es gilt die Anzeigenpreisliste ab 01.01.2017.*

**Distribution:**
IPS Pressevertrieb GmbH
Postfach 12 11, 53334 Meckenheim
Tel. 0 22 25/88 01-0
Fax 0 22 25/88 01-199

Der Herausgeber ist nicht verpflichtet, unverlangt eingesandte Manuskripte oder Geräte zurückzusenden. Auch wird für diese Gegenstände keine Haftung übernommen. Nimmt der Herausgeber einen Beitrag zur Veröffentlichung an, so erwirbt er gleichzeitig das Nachdruckrecht für alle ausländischen Ausgaben inklusive Lizenzen. Die in dieser Zeitschrift veröffentlichten Beiträge, insbesondere alle Aufsätze und Artikel sowie alle Entwürfe, Pläne, Zeichnungen einschließlich Platinen sind urheberrechtlich geschützt. Ihre auch teilweise Vervielfältigung und Verbreitung ist grundsätzlich nur mit vorheriger schriftlicher Zustimmung des Herausgebers gestattet. Die veröffentlichten Schaltungen können unter Patent- oder Gebrauchsmusterschutz stehen. Herstellen, Feilhalten, Inverkehrbringen und gewerblicher Gebrauch der Beiträge sind nur mit Zustimmung des Verlages und ggf. des Schutzrechtsinhabers zulässig. Nur der private Gebrauch ist frei. Bei den benutzten Warenbezeichnungen kann es sich um geschützte Warenzeichen handeln, die nur mit Zustimmung ihrer Inhaber warenzeichengemäß benutzt werden dürfen. Die geltenden gesetzlichen Bestimmungen hinsichtlich Bau, Erwerb und Betrieb von Sende- und Empfangseinrichtungen und der elektrischen Sicherheit sind unbedingt zu beachten. Eine Haftung des Herausgebers für die Richtigkeit und Brauchbarkeit der veröffentlichten Schaltungen und sonstigen Anordnungen sowie für die Richtigkeit des technischen Inhalts der veröffentlichten Aufsätze und sonstigen Beiträge ist ausgeschlossen.

© 2017 elektor international media b.v.
Druck: Senefelder Misset, Doetinchem (NL)
ISSN 0932-5468

# Nachbau, der sich auszahlt

Im Editorial der letzten Ausgabe habe ich Ihnen eine Smartwatch vorgestellt, die von einem jungen Maker konzipiert, entwickelt, programmiert und zusammengebaut wurde. Die intelligente und kommunikative Armbanduhr war einer der Preisträger des ersten „Fast Forward Awards (FFA)“, den Elektor zusammen mit der Messe München veranstaltet hat.

In dieser Ausgabe finden Sie zwei weitere Beiträge zu diesem Startup- und Ideen-Wettbewerb. Der Impedanz-Analyser von Wil Dijkman ist ein nützliches Messgerät, das sich nicht nur für „HomeLabs“, sondern auch für Schulen und andere Lehreinrichtungen eignet. Wer den Nachbau in Angriff nimmt, kann wie immer eine Menge lernen. Er erhält aber auch ein Gerät, das handlicher und vor allem viel kostengünstiger ausfällt als ein vergleichbares, kommerziell erhältliches Modell. Damit wird eine gute, alte Elektor-Tradition hochgehalten, was mich besonders freut.

Während beim Impedanz-Analyser eine Menge (cleverer) analoger Schaltungstechnik zum Einsatz kommt, besticht das zweite FFA-Projekt in dieser Ausgabe durch ein ausgefuchstes Bundle aus modularer Hard- und Software. Beim Sensorinterface „GoNotify“, das sich in der Hausautomatisierung und allerlei IoT-Projekten nutzen lässt, hat mich vor allem die breite Palette an Demos fasziniert, die der Autor mitliefert. Im Artikel finden Sie sechs tolle Denkanstöße, wie Sie das kleine Modul gewinnbringend einsetzen können – vom Wasserzähler mit Alarmfunktion bis zum MQTT-Client, der mit der Cloud-Plattform AllThingsTalk kommuniziert. Letztere will ich übrigens auch einmal selbst im Rahmen meines eigenen IoT-Blogs ausprobieren.

Wenn Sie auch gerade an einem (analogen oder digitalen) Projekt tüfteln, freue ich mich über eine Mail von Ihnen oder einen Eintrag auf unserer Projektplattform elektormagazine.de/labs. Noch einmal ausdrücklich will ich betonen, dass es nicht immer so ein großes Projekt sein muss wie die gerade eben geschilderten – auch die einfachen und kleinen Schaltungen sind willkommen!

Jens Nickel
Chefredakteur Elektor

## Unser Team

| | |
|---|---|
| Chefredakteur: | Jens Nickel (v.i.S.d.P.) (redaktion@elektor.de) |
| Ständige Mitarbeiter: | Dr. Thomas Scherer, Rolf Gerstendorf |
| Leserservice: | Ralf Schmiedel |
| Korrekturen: | Malte Fischer |
| Internationale Redaktion: | Thijs Beckers, Eric Bogers, Jan Buiting, Mariline Thiebaut-Brodier |
| Elektor-Labor: | Ton Giesberts, Hedwig Hennekens, Luc Lemmens, Jan Visser, Clemens Valens |
| Grafik & Layout: | Giel Dols |

# Inhalt

48. Jahrgang – Nr. 557
September/Oktober 2017

## Rubriken

3 **Impressum**

52 **Retronik**
Als die Bilder sprechen lernten

63 **HomeLab Helikopter**

66 **Elektor-Videoolympiade**
Dabei sein ist gut – gewinnen ist besser!

101 **Elektormagazine.de/Labs**
Projekte in der Pipeline

106 **Bemerkenswerte Bauteile**
Bandgap-Referenz

126 **Elektor Ethics**
Das IoT: Verantwortung für alle Beteiligten

130 **Hexadoku**

## Hintergrund

6 **ESP32 für Power-User**
Native Programmierung

25 **Raspberry Pi Zero W**
Erste Schritte mit der Mini-Himbeere

34 **Q & A**
Qualität von Audio-Steckern

44 **MiniPRO-Programmer TL866A**
Viele Möglichkeiten für wenig Geld

46 **REDEXPERT: Simulator für induktive Bauelemente**
Spulen berechnen, auswählen und Verluste kalkulieren

73 **iPhone/iPad-Anwendungen einfach entwickeln**
Externe Elektronik ansteuern über WLAN und BLE

96 **Der I²C-Bus**
Teil 3: Bausteine und Fehleranalyse

107 **Sigfox**
Welt der „Things“

112 **Digitale Korrelation**
Erkennen von Schwingungsform und Periode trotz Rauschen

# Fernbedienung mit Bluetooth Low Energy

## Mit nur einem Knopfdruck...

84

# Klein, aber oho: Raspberry Pi Zero W

## Erste Schritte mit der Mini-Himbeere

Die Mini-Version des Raspberry Pi mit dem Namenszusatz Zero gibt es bereits seit Ende 2015. Im Februar ist nun eine um WLAN und Bluetooth ergänzte Variante Zero W („Wireless“) erschienen. In diesem Artikel schauen wir uns an, mit welchen Schritten man die neue Platine an den Start bringt, was sie leistet und welche Vorteile die Funktechnologie in Verbindung mit dem winzigen Single-Board-Computer ermöglicht. In einem Praxisbeispiel schließen wir einen Temperatursensor an, und verschicken die Daten per MQTT-Protokoll über WLAN.

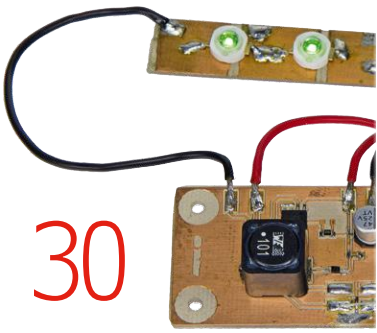
25

## Projekte

14 **Muskelsteuerung für Servo-Motoren**
Arduino liest EMG-Signale

18 **Jammer-Alarm**
Meta-Alarmanlage schützt vor Störsendern

20 **Erste Schritte mit LoRa**
Kurzstreckenfunk mit zwei Eval-Boards



30 **LED-Treiber für 12 V**

elektor magazine


**78 Kinetische Skulptur**
Treffen sich ein E-Techniker und ein Künstler...

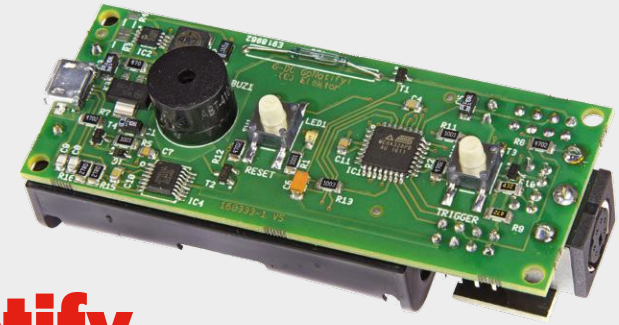
**84 Fernbedienung mit Bluetooth Low Energy**
Mit nur einem Knopfdruck...

**90 Das TABULA-Projekt**
Objekte auf Touchscreens kapazitiv erfassen

**102 Marderscheuche**
Mit unhörbaren 90 dB

**116 Impedanz-Analyser**
Eher seltenes, aber nützliches Messgerät

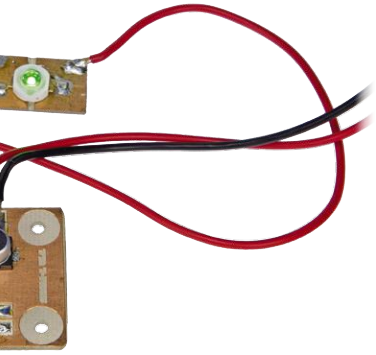

36



# GoNotify, ein flexibles IoT-Sensorinterface

## Einfach in die Cloud

Wir hören viel über IoT = das Internet der Dinge, IIoT, vernetzte Sensoren und Kühlschränke.... In ein paar Jahren soll es einen riesigen Markt und Milliarden von vernetzten Geräten geben, aber Sie gehen immer noch zu einem Wandschalter, um das Licht im Wohnzimmer anzuschalten? Jetzt sind die dunklen Tage aber vorbei, denn mit GoNotify kann man ans Internet anschließen, was einem gefällt.



**Bringt Licht in dunkle Kajüten**


**30 LED-Treiber für 12 V**
Bringt Licht in dunkle Kajüten

**36 GoNotify, ein flexibles IoT-Sensorinterface**
Einfach in die Cloud

**56 Candle 2 LED**
Kerzenlicht-Effizienzbooster

**68 FPGA/DSP-Board für Schmalband-SDR**
Teil 2: Sender und HF-Frontend


## Vorschau

Das nächste Heft ist wie immer randvoll gefüllt mit Schaltungsprojekten, Grundlagen sowie Tipps und Tricks für Elektroniker.

**Aus dem Inhalt:**

- Wetter-Anzeige auf Grafikdisplay
- Grundlagen: Akkus und Ladetechnik
- Küchenwaage mit Arduino
- Volumeneinstellung für RPi-Audio-DAC
- PlatformIO – eine IDE für viele Controller
- CAN-Bus-Debugger
- Test Funktionsgeneratoren
- 7-Segment-Display für Sport und Spiel

**Und vieles mehr!**

Änderungen vorbehalten.
Elektor November/Dezember 2017 erscheint am 18. Oktober 2017.
Verkaufsstellen findet man unter www.pressekaufen.de.

## Vorschau Elektor Business
### Internet of Things

- 5G steht vor der Tür
- Flexible IoT-Module
- Revolution Pi
- Schwarmintelligenz für Energie-Netze

**Und vieles mehr!**

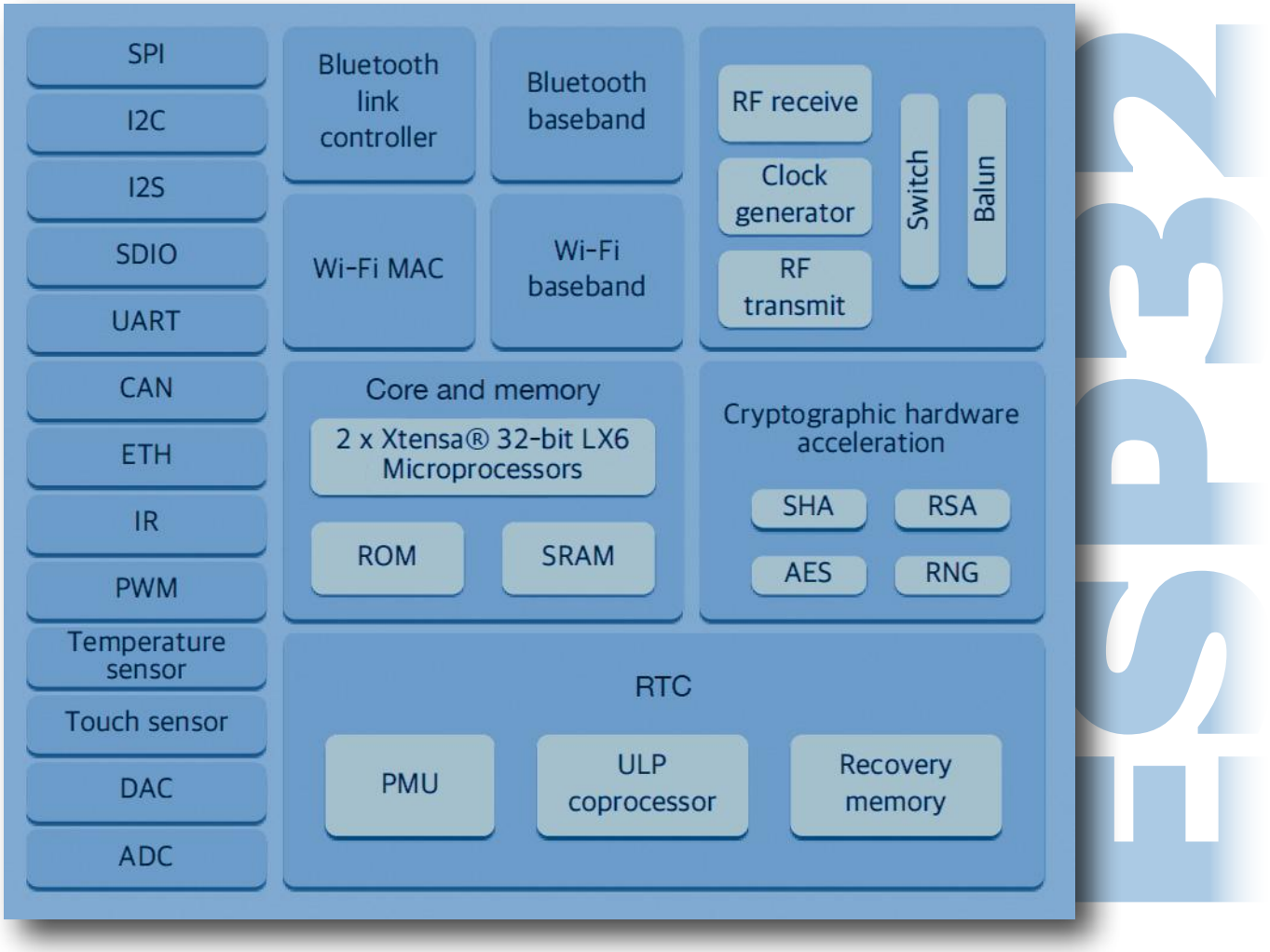
Änderungen vorbehalten.
Elektor Business 5/2017 erscheint am 15. September 2017.

# ESP32 für Power-User

## Native Programmierung

Von **Tam Hanna**

In der letzten Ausgabe haben wir gezeigt, wie einfach der mächtige ESP32 über die Arduino-IDE programmierbar ist [1]. Wer alle Funktionen des mit WLAN und Bluetooth ausgestatteten Mikrocontrollers nutzen will, muss allerdings zum nativen ESP-IDF (IoT Development Framework) greifen. Da die Arbeit mit den diversen Kommandozeilen-Werkzeugen für Quereinsteiger und Linux-unerfahrene Entwickler mitunter einschüchternd ist, zeigt dieser Artikel die Vorgehensweise.




Beim Hersteller Espressif arbeitet man größtenteils mit Ubuntu: Wir wollen die folgenden Schritte unter Ubuntu 14.04 LTS durchführen. Wer Windows verwendet, bekommt unter [2] weitere Informationen. Besitzer eines Apple werden unter [3] bedient.

Die Toolchain wird vom Hersteller in Form eines Binärpakets bereitgestellt. Zu seiner Nutzung sind einige Hilfsprogramme notwendig, die mit folgendem Kommando heruntergeladen und installiert werden können:

```
sudo apt-get install git wget make libncurses-dev
flex bison gperf python python-serial
```

Hierzu ein kleiner Hinweis: `apt-get install` stört sich nicht daran, wenn sich einige der angeforderten Pakete (siehe Parameter) schon auf dem Zielsystem befinden - sie werden in diesem Fall einfach kommentarlos übergangen.
Wer unter einem 64-Bit-Betriebssystem arbeitet, lädt das File unter [4] herunter. Besitzer eines 32-Bitters finden unter [5]

ein File, das allerdings nicht sonderlich unterstützt wird.

Beim Herunterladen mit Firefox wandern die Inhalte automatisch in den *Downloads*-Ordner. Führen Sie die folgende Kommandofolge aus, um sie in das Unterverzeichnis *esp* zu extrahieren und bereitzustellen:

```
tamhan@TAMHAN14:~$ mkdir -p ~/esp

tamhan@TAMHAN14:~$ cd ~/esp

tamhan@TAMHAN14:~/esp$ tar -xzf ~/Downloads/xtensa-esp32-elf-linux64-1.22.0-61-gab8375a-5.2.0.tar.gz
```

Für Unix-Quereinsteiger sei die Rolle der Tilde an dieser Stelle kurz erklärt: Es handelt sich dabei um ein einzelnes Zeichen, das den Pfad des Heim-Ordners des gerade angemeldeten Nutzers beschreibt und so Vertipper verhindert.
Die ESP32-Toolchain erwartet, dass die Variable PATH ein ganz bestimmtes Verzeichnis enthält. Dies lässt sich durch Eingeben des Export-Befehls bewerkstelligen – achten Sie darauf, dass seine Wirkung mit dem Schließen des Konsolenfensters endet:

```
tamhan@TAMHAN14:~$ export PATH=$PATH:$HOME/esp/xtensa-esp32-elf/bin
```

Zu guter Letzt müssen wir die eigentliche Unterstützungsbibliothek aus GitHub herunterladen. Achten Sie darauf, das Kommando im Heimverzeichnis der ESP-Toolchain anzustoßen:

```
tamhan@TAMHAN14:~/esp$ git clone --recursive https://github.com/espressif/esp-idf.git
```

### Ein erstes Projekt

Wer ein neues Projekt erstellen möchte, muss zuerst GitHub anzapfen. Der Prozessorhersteller stellt ein Template bereit, das sich folgendermaßen herunterladen lässt:

```
tamhan@TAMHAN14:~/esp$ git clone https://github.com/espressif/esp-idf-template.git elektor1
```

Das `clone`-Kommando nimmt als Parameter den Namen des Ordners entgegen, in dem die Projektstruktur errichtet werden soll.
Dort ignorieren wir zuerst einmal die diversen Makefiles und wenden uns der Datei *main.c* zu, die im gleichnamigen Unterordner liegt. Espressif stattet sie mit einem vergleichsweise komplexen Skelett aus, das wir an dieser Stelle aber nicht weiter verwenden wollen. Wir wollen stattdessen die schon im vorigen Artikel als Demo genutzte Sägezahn-Welle ausgeben, weshalb wir den Inhalt von *main.c* ausräumen müssen (Download aller Codebeispiele unter [6]).
Im ersten Schritt ist - wie immer - die Inklusion diverser Headerfiles erforderlich, die die API bereitstellen. An dieser Stelle fällt auf, dass einer der Header auf FreeRTOS hinweist – Espressif nutzt das Echtzeitbetriebssystem an diversen Stellen:

```
#include “freertos/FreeRTOS.h”
#include “esp_system.h”
#include “esp_event.h”
```

**Make-was?**

In der Unix-Welt hat sich das Kommandozeilenwerkzeug *Make* als Quasistandard für die Automatisierung von Buildprozessen (Kompilierung, Linken usw.) etabliert. Die zur Steuerung des Kompilationsprozesses vorgesehenen Anweisungsfiles werden als *Makefile* bezeichnet.

```
#include “esp_event_loop.h”
#include “nvs_flash.h”
#include “driver/gpio.h”
#include <driver/dac.h>
```

Für die Eventschleife des Echtzeitbetriebssystems ist ein Eventhandler erforderlich, der bei uns aber immer OK retourniert und die Ausführung so nicht weiter beeinflusst:

```
esp_err_t event_handler(void *ctx, system_event_t *event) {
    return ESP_OK;
}
```

Darauf folgt auch schon die Funktion `app_main`, die beim Start des Controllerprogramms aufgerufen wird. Sie beginnt mit der Initialisierung des externen Flashspeichers und dem Einschreiben des Eventhandlers:

```
void app_main(void) {
    nvs_flash_init();
    ESP_ERROR_CHECK( esp_event_loop_init(event_handler, NULL) );
```

Im nächsten Schritt können wir den im letzten Teil der Serie verwendeten Code portieren. Die unter [7] bereitstehende Dokumentation zeigt, dass nur eine Methode erforderlich ist:

```
    while (true) {
    for(int i=0;i<255;i++){
                dac_out_voltage(DAC_CHANNEL_1, i);
        }
    }
}
```

### Finde das Board

In der Arduino-IDE wären wir an dieser Stelle fertig: Ein Klick auf Run, und der Digitalspeicheroszillograph hat Dienst. Bei der Arbeit mit Kommandozeilenwerkzeugen ist dagegen etwas Handarbeit erforderlich.

Als erste Aufgabe präsentiert sich die Lokalisierung des ESP32-Boards: Unix hängt den seriellen Port des von uns benutzten Moduls ESP32 Thing (siehe [1]) irgendwo unter */dev* ein. Zwecks einfacherem Finden des Zielgeräts ist es empfehlenswert, nach dem Anstecken den Inhalt des Kernel-Logs auszuwerten. Es handelt sich dabei um eine Art Ringspeicher, der vom Linux-Kernel während der Arbeit des Systems mit diversen Informationen bevölkert wird.

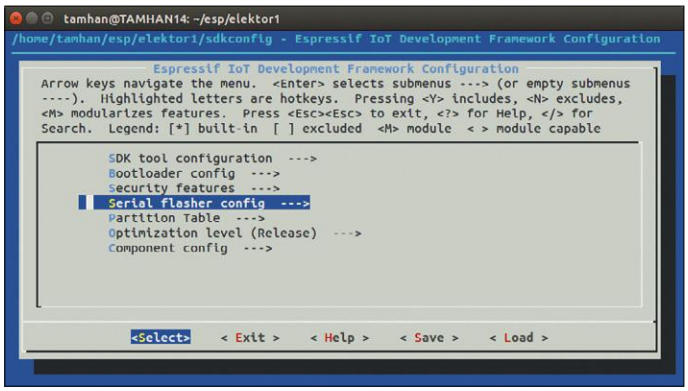



Bild 1. Der Homescreen von *make menuconfig* wartet auf Eingaben.

Das hier gezeigte Kommando leitet die Ausgabe von *dmesg* in das Tool *grep* um, das den in der Ausgabe des FTDI-Treiber befindlichen String sucht und nur die relevanten Zeilen ausgibt:

```
tamhan@TAMHAN14:~/esp$ dmesg | grep 'FTDI USB Serial Device converter now attached'

[    4.817153] usb 1-1.6: FTDI USB Serial Device converter now attached to ttyUSB0
```

Wichtig ist hier der Wert, der beschreibt, an welcher Stelle das neue Gerät im Gerätebaum eingepflanzt wurde. In unserem Fall handelt es sich hierbei um *ttyUSB0*, der Pfad lautet also */dev/ttyUSB0*.
Da der Zugriff auf serielle Geräte normalerweise dem Superuser vorbehalten ist, wollen wir unser Linux-Benutzerkonto zum Zugriff auf den Port befähigen. Dazu müssen wir im ersten Schritt feststellen, welche Nutzergruppe das Gerät besitzt. Dies lässt sich über eine Spezialvariante des `ls`-Befehls bewerkstelligen, die zusätzliche Informationen zu einem Verzeichnis oder einem angefragten Element ausgibt:

```
tamhan@TAMHAN14:~/esp/nmgsample1$ ls -l /dev/ttyUSB0

crw-rw---- 1 root dialout 188, 0 feb 26 22:58 /dev/ttyUSB0
```

Unter Unix handelt es sich normalerweise um *dialout*. Im nächsten Schritt muss das verwendete Benutzerkonto zu dieser Gruppe hinzugefügt werden, um die Zugriffsrechte zu gewähren:

```
root@TAMHAN14:~/esp/nmgsample1# sudo adduser tamhan dialout

Adding user `tamhan' to group `dialout' ...

Adding user tamhan to group dialout

Done.

root@TAMHAN14:~/esp/nmgsample1# sudo reboot
```

Unix aktualisiert die Zugriffsberechtigungen im Rahmen eines Neustarts. Wer sich also das permanente voranstellen von `sudo` ersparen möchte, muss seine Workstation an dieser Stelle neu starten.

## Menuconfig

Im nächsten Schritt können wir uns der Konfiguration der Ausführungsumgebung zuwenden. Hierbei wird - wie so oft - das *make*-Werkzeug eingespannt, das für die Abarbeitung von Kompilationsanweisungen zuständig ist.
Da das manuelle Editieren von Makefiles eine undankbare Arbeit ist, gibt es mit *menuconfig* ein mehr oder weniger stark standardisiertes Bearbeitungswerkzeug. Menuconfig kommt nicht nur im ESP-IDF zum Einsatz, sondern wird auch bei der Kompilation von Kernels und Betriebssystemimages (z.B. für OpenWRT) eingesetzt.
Die menuconfig-Variante des ESP-IDF erwartet eine Variable namens `IDF_PATH`, die auf das Verzeichnis zeigt, in dem der

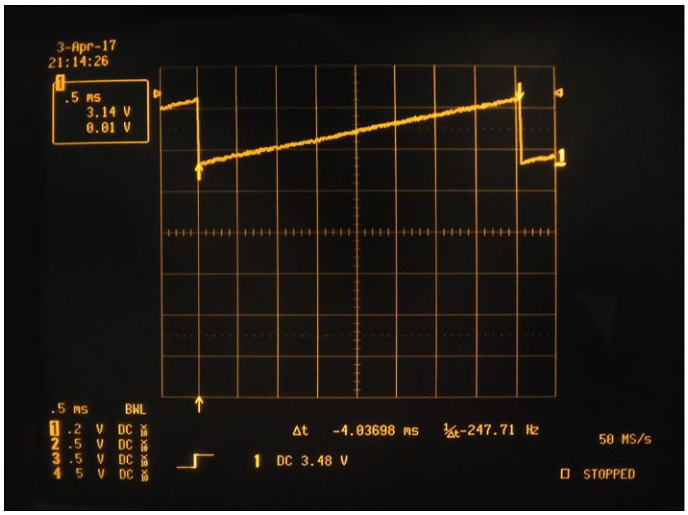



Bild 2. Das ESP-IDF ist hier langsamer als die Arduino-IDE.

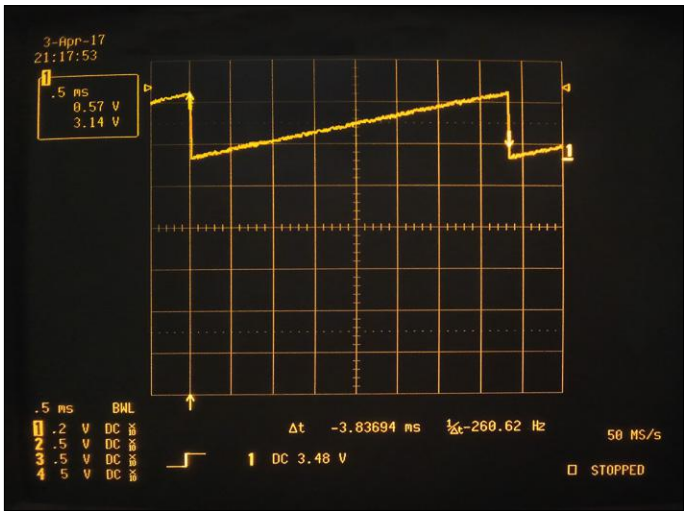



Bild 3. Der Wechsel in den Release-Modus bringt nicht sonderlich viel.

Hauptteil des Bibliothekscodes liegt. Damit sieht der Aufruf folgendermaßen aus:

```
tamhan@TAMHAN14:~/esp/elektor1$ export IDF_PATH=~/
esp/esp-idf

tamhan@TAMHAN14:~/esp/elektor1$ make menuconfig
```

Wie im weiter oben erwähnten Fall der Variable `PATH` gilt auch hier, dass die Export-Zuweisung nur so lange gültig ist, wie das Terminal-Fenster, das von ihr beeinflusst wird, geöffnet bleibt. Wundern Sie sich nicht, wenn während der Abarbeitung von `make menuconfig` Meldungen eines Compilers am Bildschirm erscheinen – Teile des Werkzeugs werden direkt vor dem Einsatz kompiliert. Die eigentliche Bedienung erfolgt sodann über die Cursortasten und die Enter-Taste, die zum Bestätigen von Auswahloptionen dient (**Bild 1**).

Die erste wichtige Einstellung findet sich unter *Serial flasher config* ➞ *Default serial port*. Geben Sie dort den weiter oben ermittelten Port ein, um die Toolchain mit dem an den PC angeschlossenen ESP32 zu verbinden. Achten Sie darauf, die Konfiguration sodann mittels *Save* zu speichern.
Nach getaner Arbeit müssen Sie die Firmware auf den Controller hochladen. Dies erfolgt ebenfalls über `make`, der Befehl lautet nun jedoch folgendermaßen:

```
tamhan@TAMHAN14:~/esp/elektor1$ make flash

GENCONFIG

CC src/bootloader_flash.o

. . .

Wrote 16384 bytes at 0x00008000 in 1.4 seconds (91.0
kbit/s)...
Hash of data verified.
```

Bild 4. Auch ohne die FreeRTOS-Eventschleife wird der Code nur unwesentlich schneller.

```
Leaving...
Hard resetting...
```

Beachten Sie, dass sich die Ausgabe des Kommandos je nach Betriebszustand unterscheidet: Während nach der ersten Kompilation eine vollständige Abarbeitung der gesamten Bibliothek erforderlich ist, laufen spätere Kompilationen schneller ab.

## Und läuft …

Wer einen Speicheroszillographen mit Pin 25 des ESP32 verbindet, sieht das in **Bild 2** gezeigte und nur wenig befriedigende Resultat. Es ist offensichtlich, dass die native Version des Codes wesentlich langsamer ist als die im letzten Heft vorgestellte Arduino-Variante.
Kehren Sie sodann in *menuconfig* zurück, um die diversen Einstellungen zu adjustieren. Als erstes setzen wir den Wert *Optimization Level* auf *Release*, und deployen das Programm nach dem Speichern der Änderungen abermals. Der Lohn der Mühen ist etwa 10 % mehr Geschwindigkeit (**Bild 3**). Eine native API ist also nicht immer effizienter als die Arduino-API.

Als Nächstes wollen wir die im Hintergrund laufende Event-Loop deaktivieren. Dazu kommentieren wir eine Zeile in `app_main` aus:

```
void app_main(void) {
    nvs_flash_init();

    //ESP_ERROR_CHECK( esp_event_loop_init(event_
handler, NULL) );
```

An dieser Stelle stellen wir fest, dass es abermals zu einer - wenn auch nur leichten - Geschwindigkeitssteigerung kommt (siehe **Bild 4**). Trotz alledem gilt, dass wir noch nicht auf einem mit dem Arduino-Sketch vergleichbaren Geschwindigkeitsniveau angekommen sind. Die Ursache dafür ist das Echtzeitbetriebssystem, das vor jedem Beschreiben der für die Ausgabe zuständigen Register einige teure Synchronisationsoperationen durchführt.

## Blauzahn

Unsere nächste Aufgabe ist das Aufbauen einer Funkverbindung zu einem Smartphone: Die wichtigste neue Funktion des ESP32 ist der Bluetooth-Transceiver, den wir nun ein wenig in Aktion bringen wollen. Zuerst müssen wir das Bluetoothmodul des Controllers aktivieren: Öffnen Sie dazu *make menuconfig*, und wechseln Sie in die Rubrik *Component Config*. Aktivieren Sie sodann Bluetooth durch Drücken von Y – eine Option ist aktiviert, wenn ein * in den beiden eckigen Klammern aufscheint. Speichern Sie die Konfiguration sodann; im Rahmen der nächsten Kompilation werden einige Dutzend zusätzliche Dateien kompiliert.
Espressif setzt auf den Bluedroid-Stack: Wer viel Erfahrung mit Android mitbringt, dürfte die eine oder andere Methode bereits kennen. Für alle anderen gibt es hier eine Kurzeinführung. Bevor wir in medias res gehen, sei allerdings noch eine allgemeine Anmerkung erlaubt.
Bei der Arbeit mit komplexen Softwaresystemen – Echtzeitbetriebssysteme wie FreeRTOS gehören definitiv dazu – ist es in höchstem Maße empfehlenswert, bei der Erzeugung eigener Programme nicht mit einem „clean sheet“ anzufangen.

Schnappen Sie sich stattdessen ein Codebeispiel, das eine ähnliche Funktionalität realisiert – analysieren Sie seine Struktur, und adaptieren beziehungsweise übernehmen Sie Elemente Schritt für Schritt.
Der – um das Logging von Fehlern gekürzte – Einsprungpunkt unseres Programms sieht so aus:

```
void app_main() {
    esp_err_t ret;
    esp_bt_controller_config_t bt_cfg =
BT_CONTROLLER_INIT_CONFIG_DEFAULT();
    ret = esp_bt_controller_init(&bt_cfg);

    . . .
```

Als Erstes rufen wir die Funktion `esp_bt_controller_init` auf, die für die Initialisierung des Bluetooth-Subsystems als Ganzes verantwortlich ist. Das übergebene Konfigurationsobjekt wird über ein Makro beschafft, das eine „Standardimplementierung" der Bluetooth-Struktur zurückgibt.

Im nächsten Schritt folgt die Definition des Betriebsmodus des Bluetooth-Controllers: Wir nutzen hier BTDM, um sowohl BT LE als auch Classic zu aktivieren. Danach werden noch zwei Housekeeping-Funktionen aufgerufen, die dem Bluetooth-Stack Speicher und Rechenzeit zuweisen:

```
    ret = esp_bt_controller_enable(ESP_BT_MODE_BTDM);

    . . .

    ret = esp_bluedroid_init();

    . . .

    ret = esp_bluedroid_enable();

    . . .
```

Die Bluetooth-LE-Implementierung des ESP32 arbeitet hochgradig asynchron. Aus diesem Grund schreiben wir zwei Eventhandler ein, die für GATT- und GAP-Ereignisse [8] verantwortlich sind. Zu guter Letzt wird beim GATT-Protokoll noch eine Applikation angemeldet, in der später Attribute eingeschrieben werden können:

```
    esp_ble_gatts_register_callback(gatts_event_
handler);
    esp_ble_gap_register_callback(gap_event_handler);

    esp_ble_gatts_app_register(0); //App-ID 0
    return;
}
```

Für die Nutzung der Bluetooth-APIs ist zudem die Inklusion einer Gruppe von Headern erforderlich. Warum Espressif hier kein „Catch-All-File" anbietet, das alle Header in einem Schritt bereitstellt, ist dem Autor unverständlich:

```
#include "esp_system.h"
#include "esp_log.h"
#include "nvs_flash.h"
#include "bt.h"
#include "bta_api.h"
#include "esp_gap_ble_api.h"
#include "esp_gatts_api.h"
#include "esp_bt_defs.h"
#include "esp_bt_main.h"
#include "esp_bt_main.h"
#include "sdkconfig.h"
```

### Handler herbei

Bluetooth LE wurde von Anfang an als stromsparendes Kommunikationssystem entwickelt: die Nutzung von „synchronen" Methoden und Polling negiert diese Vorteile durch eine höhere Belastung des Hauptprozessors. Aus diesem Aspekt ist es nicht verwunderlich, dass der Bluetooth-LE-Stack komplett asynchron aufgebaut ist und das Einschreiben von Ereignishandlern voraussetzt.

`gap_event_handler` ist für Ereignisse des GAP-Protokolls verantwortlich. Seine Hauptaufgabe ist das Aussenden von als Advertisement bekannten Paketen, die andere Hardware über das Vorhandensein des Geräts informieren. Der Aufruf von `esp_ble_gap_start_advertising` weist den Stack hierbei dazu an, einen neuen Werbezyklus zu starten:

```
static void gap_event_handler(esp_gap_ble_cb_event_t
event, esp_ble_gap_cb_param_t *param)
{

    switch (event) {

    case ESP_GAP_BLE_ADV_DATA_SET_COMPLETE_EVT:
        esp_ble_gap_start_advertising(&test_adv_
params);
        break;

    case ESP_GAP_BLE_ADV_DATA_RAW_SET_COMPLETE_EVT:
        esp_ble_gap_start_advertising(&test_adv_
params);
        break;
```

Die diversen Parameter des Werbe-Kommandos werden in Form einer `esp_ble_adv_params_t`-Struktur angeliefert, deren Einstellungen 1:1 von der Vorlage übernommen werden dürfen:

```
static esp_ble_adv_params_t test_adv_params = {
    .adv_int_min        = 0x20,
    .adv_int_max        = 0x40,
    .adv_type           = ADV_TYPE_IND,
    .own_addr_type      = BLE_ADDR_TYPE_PUBLIC,
    .channel_map        = ADV_CHNL_ALL,
    .adv_filter_policy =
ADV_FILTER_ALLOW_SCAN_ANY_CON_ANY,
};
```

Interessanterweise setzt die Nutzung des Stacks die Einbindung der Datei *string.h* voraus, die folgendermaßen erfolgt:

```
#include <string.h>
```

Als Nächstes ist die Deklaration von Speicherbereichen erforderlich, in denen der Bluetooth-Stack temporäre Informationen ablegen kann. Zudem entsteht eine Struktur vom Typ `esp_attr_value_t`, die das anzulegende Attribut beschreibt:

```
#define GATTS_DEMO_CHAR_VAL_LEN_MAX 0x40

uint8_t char1_str[] = {0x11,0x22,0x33};

esp_attr_value_t gatts_demo_char1_val =
{
    .attr_max_len = GATTS_DEMO_CHAR_VAL_LEN_MAX,
    .attr_len     = sizeof(char1_str),
    .attr_value   = char1_str,
};
```

Bluetooth LE identifiziert Geräte und Gegenstellen unter Nutzung von als UUID bezeichneten numerischen IDs, die schon ob ihrer immensen Länge – im Allgemeinen – weltweit einzigartig sind. Die Bluetooth-API unterstützt Entwickler durch den Typ `esp_bt_uuid_t` beim Hantieren mit ebendiesen:

```
esp_gatt_srvc_id_t service_id;
uint16_t service_handle;
esp_bt_uuid_t descr_uuid;
esp_bt_uuid_t char_uuid;
```

Aufmerksame Leser fragen an dieser Stelle, wieso zwei Charakteristiken zur Realisierung unseres kleinen Diensts erforderlich sind. Wir legen hier neben der eigentlichen Charakteristik auch einen Descriptor an, der Informationen über die in der Charakteristik enthaltenen Daten bereitstellt.
Der eigentliche Eventhandler ist dann für das Beantworten der diversen in Bluetooth LE auftretenden Ereignisse verantwortlich. Wir beschränken uns aus Gründen der Kompaktheit hier auf die Realisierung einer Nur-Lese-Charakteristik, was zu folgendem Code führt:

```
static void gatts_event_handler(esp_gatts_cb_event_t
event, esp_gatt_if_t gatts_if, esp_ble_gatts_cb_
param_t *param)
{
    switch (event) {

    case ESP_GATTS_REG_EVT:
        ESP_LOGI(GATTS_TAG, “REGISTER_APP_EVT, status
%d, app_id %d\n”, param->reg.status, param->reg.
app_id);
        service_id.is_primary = true;
        service_id.id.inst_id = 0x00;
        service_id.id.uuid.len = ESP_UUID_LEN_16;
        service_id.id.uuid.uuid.uuid16 =
GATTS_SERVICE_UUID_TEST_A;
        esp_ble_gap_set_device_name(“ElektorTest”);
```

**Aktuelle Bluetooth-Bibliothek notwendig!**

Espressif hat die Bluetooth-API in den letzten Monaten mehrfach stark überarbeitet. Wenn das hier gezeigte Beispiel nicht funktioniert, so müssen Sie Ihre IDF-Installation aktualisieren. Dies erfolgt am einfachsten durch Löschen und Neu-Herunterladen der Bibliotheken. Achten Sie darauf, die Konfiguration danach durch einen weiteren Aufruf von `make menuconfig` zu aktualisieren und das „neue“ File zu speichern.

**Mehr Bluetooth-Beispiele**

Espressif bietet eine Vielzahl von Beispielen zu den diversen Bluetooth-Betriebsmodi an – weitere Informationen finden sich unter [10].

`ESP_GATTS_REG_EVT` ist hierbei für die Anmeldung einer neuen Charakteristik beim Stack verantwortlich: Der ServiceID-Parameter wird im Rahmen der Abarbeitung „bevölkert“. `CREATE_EVT` wird dann aufgerufen, wenn der Korpus erzeugt wurde und zur Parametrierung bereitsteht:

```
  case ESP_GATTS_CREATE_EVT:
      service_handle = param->create.service_handle;
      char_uuid.len = ESP_UUID_LEN_16;
      char_uuid.uuid.uuid16 = 0xFF01;
      esp_ble_gatts_start_service(service_handle);
      esp_ble_gatts_add_char(service_handle,
      &char_uuid, ESP_GATT_PERM_READ |
      SP_GATT_PERM_WRITE,
      ESP_GATT_CHAR_PROP_BIT_READ |
      ESP_GATT_CHAR_PROP_BIT_WRITE |
      ESP_GATT_CHAR_PROP_BIT_NOTIFY,
      &gatts_demo_char1_val, NULL);
          break;
```

Auf die Charakteristik bezogene Leseanfragen werden durch Nutzung der Funktion `esp_ble_gatts_send_response` beantwortet – sie nimmt ein Bitfeld entgegen, das die in Richtung des Anfragenden zu jagenden Informationen enthält:

```
        case ESP_GATTS_READ_EVT: {
            esp_gatt_rsp_t rsp;
            memset(&rsp, 0, sizeof(esp_gatt_rsp_t));
            rsp.attr_value.handle = param->read.
handle;
            rsp.attr_value.len = 4;
            rsp.attr_value.value[0] = 0xde;
            rsp.attr_value.value[1] = 0xed;
            rsp.attr_value.value[2] = 0xbe;
            rsp.attr_value.value[3] = 0xef;
            esp_ble_gatts_send_response(gatts_if,
param->read.conn_id, param->read.trans_id, ESP_GATT_
OK, &rsp);
            }
```

```
            break;
```

ESP_GATTS_ADD_CHAR_EVT ist dann für das eigentliche Anlegen der Charakteristik verantwortlich, die weiter oben durch Aufrufen der Methode esp_ble_gatts_add_char ins Leben gerufen wurde:

```
        case ESP_GATTS_ADD_CHAR_EVT: {
            uint16_t length = 0;
            const uint8_t *prf_char;
            //gl_profile_tab[PROFILE_A_APP_ID].char_
handle = param->add_char.attr_handle;
            descr_uuid.len = ESP_UUID_LEN_16;
            descr_uuid.uuid.uuid16 =
ESP_GATT_UUID_CHAR_CLIENT_CONFIG;
            esp_ble_gatts_get_attr_value(param->add_
char.attr_handle,  &length, &prf_char);
            esp_ble_gatts_add_char_descr(service_
handle, &descr_uuid, ESP_GATT_PERM_READ | ESP_GATT_
PERM_WRITE, NULL, NULL);
            }
            break;
```

Zu guter Letzt müssen wir nach dem Aufbrechen der Verbindung zwischen Gerät und Datenquelle den Werbeprozess neu starten, um für weitere Clients sichtbar zu sein:

```
        case ESP_GATTS_DISCONNECT_EVT:
            esp_ble_gap_start_advertising(&test_adv_
params);
            break;

        default:
            break;
        }
}
```

Aus Platzgründen können wir an dieser Stelle nicht weiter auf die eigentliche Interaktion mit dem Smartphone eingehen – **Bild 5** zeigt, wie sich die erzeugte Charakteristik in der *Nordic BLE* App präsentiert. Das im Play Store unter [9] zum Download bereit stehende Programm arbeitet als eine Art „Scanner", der die Inhalte von Bluetooth-LE-Geräten analysiert und die Interaktion mit ihnen ermöglicht.

## Fazit

Auch wenn die Performance der ESP-IDF-API stellenweise nicht mit der Arduino-API mithalten kann: Wer die maximale Leistung des ESP32 in Anspruch nehmen möchte, kommt um die Beschäftigung mit dem IDF nicht herum. Wer Vorkenntnisse mit anderen 32-bit-Controllern mitbringt, kann sich in die API im Allgemeinen schnell einarbeiten. ◀

(160457)

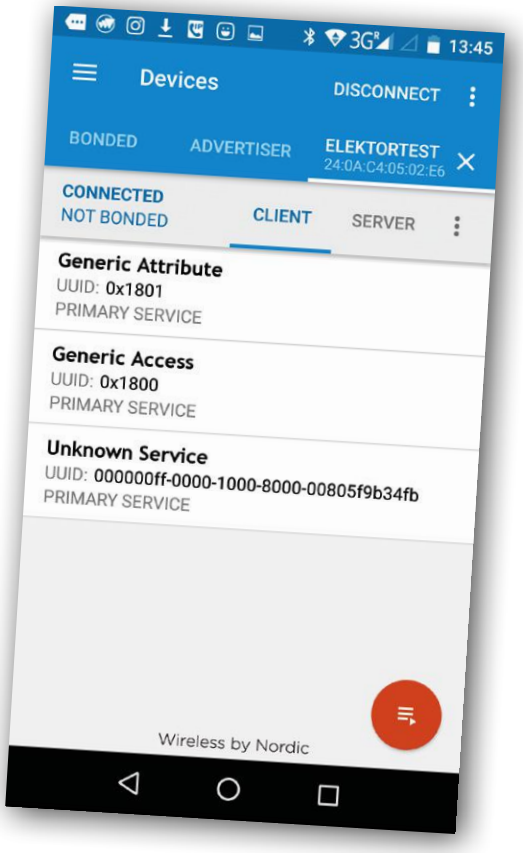



Bild 5. Die vom ESP32 angelegte Bluetooth-Charakteristik steht zum Zugriff bereit.

## Weblinks


[1] www.elektormagazine.de/160454
[2] https://esp-idf.readthedocs.io/en/latest/get-started/windows-setup.html
[3] https://esp-idf.readthedocs.io/en/latest/get-started/macos-setup.html
[4] https://dl.espressif.com/dl/xtensa-esp32-elf-linux64-1.22.0-61-gab8375a-5.2.0.tar.gz
[5] https://dl.espressif.com/dl/xtensa-esp32-elf-linux32-1.22.0-61-gab8375a-5.2.0.tar.gz
[6] www.elektormagazine.de/160457
[7] http://esp-idf.readthedocs.io/en/latest/api-reference/peripherals/dac.html
[8] https://en.wikipedia.org/wiki/List_of_Bluetooth_profiles
[9] https://play.google.com/store/apps/details?id=no.nordicsemi.android.mcp&hl=en
[10] https://github.com/espressif/esp-idf/tree/07b61d5/examples/bluetooth

# Die Elektor-Community

LEARN DESIGN SHARE

| 82 | 248153 | 1040 | 489 | 235332 |
|---|---|---|---|---|
| Länder | Mitglieder | Experten & Autoren | Literatur | Monatliche Besucher |

**Elektor durchbricht die Schranken einer Zeitschrift und wird zur Community aktiver E-Ingenieure - vom Anfänger bis zum Profi – begierig, überraschende Elektronik zu lernen, zu entwickeln, zu teilen.**

**Elektor-Shop:** 24 Stunden an 7 Tagen der Woche für jeden Elektroniker geöffnet! Dauerhafter Rabatt von 10% für alle GOLD- und GREEN-Mitglieder. **www.elektor.de**

**Elektor-Zeitschrift:** 12 Ausgaben (6 Doppelhefte + 6 Business-Ausgaben) pro Jahr voll gepackt mit Elektronik-Projekten, Artikeln, Besprechungen, Tipps und Tricks. **www.elektormagazine.de**

**Elektor-Platinen-Service:** Bestellung von Platinen als Einzelstück oder Kleinserie. **www.elektorpcbservice.de**

**Elektor wöchentlich & papierlos:** Wöchentlicher digitaler Newsletter. Kostenlos und aktuell. **www.elektor.de/newsletter**

**Elektor Academy:** Webinare, Seminare, Präsentationen, Workshops, DVDs und mehr = praxisorientiertes Lernen. **www.elektor.de**

**Elektor-Bücher & DVDs:** Arduino, Raspberry Pi, Mikrocontroller und vieles andere mehr. Im Online-Shop mit 10% Rabatt für Mitglieder! **www.elektor.de/bucher**

**Elektor.TV:** Reviews, Eindrücke, Unboxings und persönliche Journale. Anschauen heißt Erfahrung sammeln. **www.elektor.tv**

**Elektor-Labs:** Eigene Projekte vorstellen – von Anderen lernen – Anderen helfen und mit Anderen teilen. Elektor macht mit und testet Ihre Ideen! **www.elektormagazine.de/labs**

## Treten Sie dem weltweit größten Elektroniker-Netzwerk bei!

### GREEN 1,78 €/Woche

- ✔ Zugang zum Elektor-Archiv
- ✔ 10% Rabatt auf Shop-Produkte
- ✔ 6x Elektor-Doppelheft (Digital)
- ✔ 6x Elektor Business (Digital)
- ✘ 6x Elektor-Doppelheft (Print)
- ✘ 6x Elektor Business (Print)
- ✔ Exklusive Top-Angebote
- ✘ Elektor Jahrgangs-DVD

Unsere Empfehlung

www.elektor.de/green-mitglied

### GOLD 2,45 €/Woche

- ✔ Zugang zum Elektor-Archiv
- ✔ 10% Rabatt auf Shop-Produkte
- ✔ 6x Elektor-Doppelheft (Digital)
- ✔ 6x Elektor Business (Digital)
- ✔ 6x Elektor-Doppelheft (Print)
- ✔ 6x Elektor Business (Print)
- ✔ Exklusive Top-Angebote
- ✔ Elektor Jahrgangs-DVD

www.elektor.de/gold-mitglied

### GRATIS

- ✘ Zugang zum Elektor-Archiv
- ✘ 10% Rabatt auf Shop-Produkte
- ✘ 6x Elektor-Doppelheft (Digital)
- ✘ 6x Elektor Business (Digital)
- ✘ 6x Elektor-Doppelheft (Print)
- ✘ 6x Elektor Business (Print)
- ✔ Exklusive Top-Angebote
- ✘ Elektor Jahrgangs-DVD

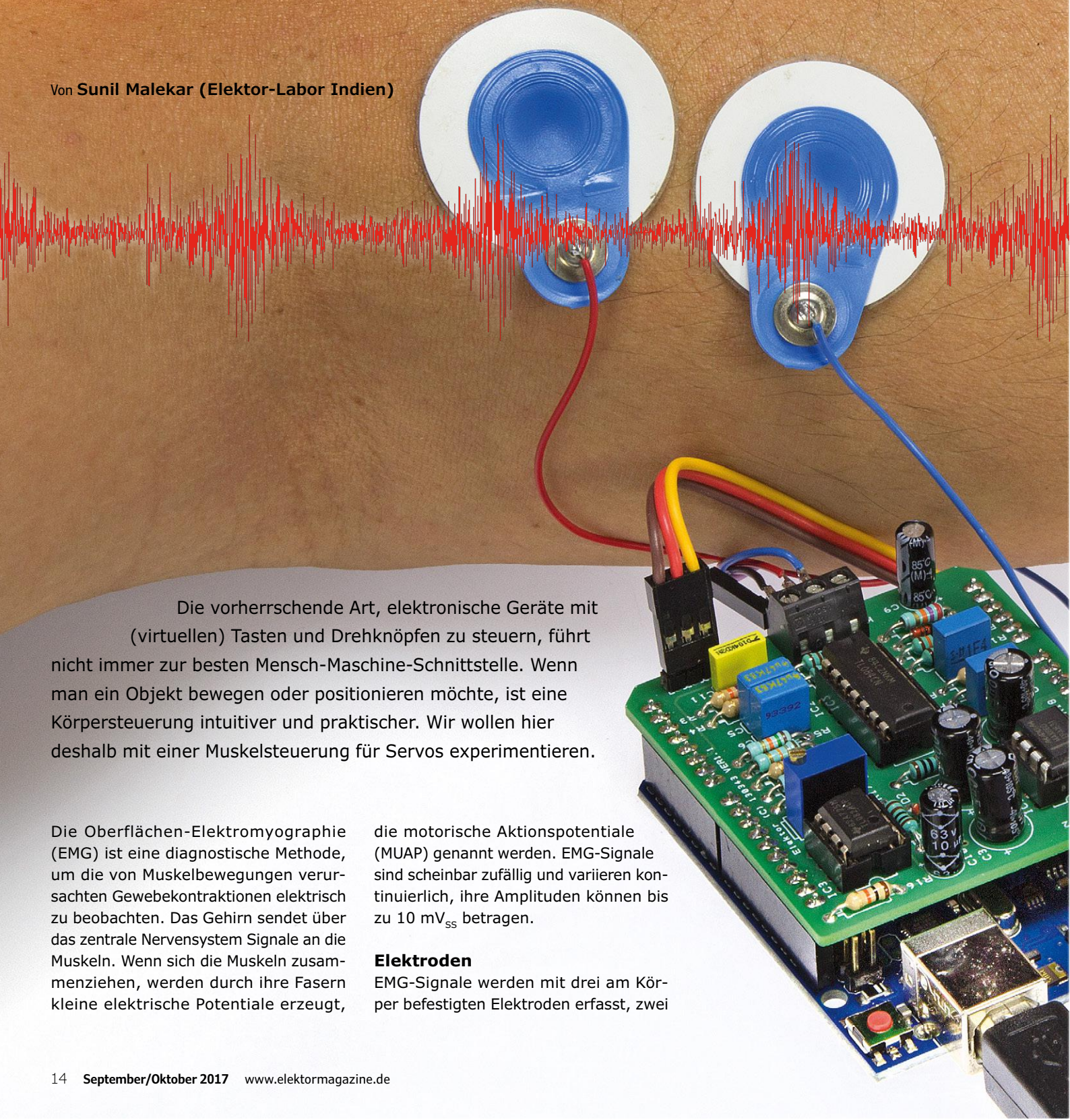
www.elektor.de/newsletter

facebook.com/ElektorDE

twitter.com/ElektorDE

# Muskelsteuerung für Servo-Motoren

Von **Sunil Malekar (Elektor-Labor Indien)**



Die vorherrschende Art, elektronische Geräte mit (virtuellen) Tasten und Drehknöpfen zu steuern, führt nicht immer zur besten Mensch-Maschine-Schnittstelle. Wenn man ein Objekt bewegen oder positionieren möchte, ist eine Körpersteuerung intuitiver und praktischer. Wir wollen hier deshalb mit einer Muskelsteuerung für Servos experimentieren.

Die Oberflächen-Elektromyographie (EMG) ist eine diagnostische Methode, um die von Muskelbewegungen verursachten Gewebekontraktionen elektrisch zu beobachten. Das Gehirn sendet über das zentrale Nervensystem Signale an die Muskeln. Wenn sich die Muskeln zusammenziehen, werden durch ihre Fasern kleine elektrische Potentiale erzeugt, die motorische Aktionspotentiale (MUAP) genannt werden. EMG-Signale sind scheinbar zufällig und variieren kontinuierlich, ihre Amplituden können bis zu 10 $mV_{SS}$ betragen.

## Elektroden

EMG-Signale werden mit drei am Körper befestigten Elektroden erfasst, zwei

davon 2...3 cm voneinander entfernt an der Stelle, wo eine Muskelbewegung erkannt werden soll. Die dritte Elektrode soll einen Referenzpegel für das zu messende EMG-Signal liefern und wird an einem elektrisch neutralen Gewebe (wo der Knochen direkt unter der Haut liegt), aber nicht allzu weit von den anderen beiden Elektroden entfernt angebracht. Zu nahe an den ersten beiden sollte sich die dritte Elektrode aber auch nicht befinden, um eine Beeinträchtigung der Signalstärke zu vermeiden. Geeignete Elektroden finden Sie leicht im Internet - suchen Sie nach EMG-Oberflächen-Klebeelektroden. Sie kosten je nach Ausführung 1...2 € pro Stück und werden in Packungen von 12 bis 50 Stück geliefert. Es gibt trockene und gelbeschichtete Elektroden.

## Eigenschaften

- Steuert Servos durch Muskelbewegungen
- Arduino-Shield
- Nur Durchsteckbauteile

# Arduino liest EMG-Signale

Wir haben letztere verwendet, auf denen eine Elektrolytlösung oder ein Gel auf der Hautseite der Elektrode aufgetragen ist. Auf der anderen Elektrodenseite ist das Kabel angebracht. An der Grenzfläche zwischen dem Gel und der Elektrode tritt eine chemische Reaktion auf, wodurch sich eine Potentialdifferenz entwickelt. Silberchlorid-Elektroden (AgCl) sind die besten, weil Impedanzschwankungen zwischen Haut und Elektrode und damit das Rauschen minimiert werden.

### Schaltungsbeschreibung

Das von den Elektroden eingefangene EMG-Signal ist schwach und muss verstärkt werden. Normalerweise wird dafür ein Instrumentenverstärker mit hoch-

## INFOS ZUM PROJEKT

Arduino
EMG
Biosteuerung
Servo

➡ Einsteiger
Fortgeschrittener
Experte

Etwa 2 Stunden

Arduino,
Lötkolben

etwa 30 €

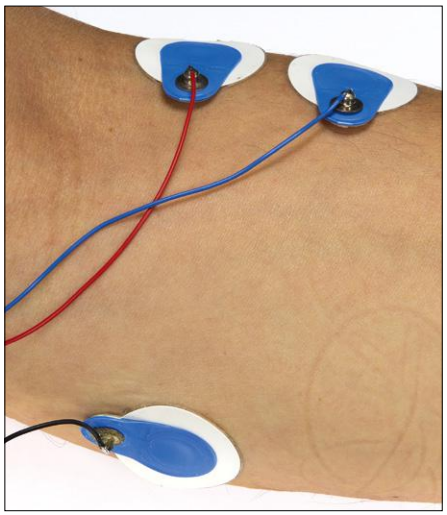


Bild 1. Geeignete Elektroden und leitendes Gel sind im Internet leicht erhältlich.

ohmigem Eingang eingesetzt, der allerdings auch Störungen (Netzbrummen, HF-Einstreuungen) munter mitverstärkt, besonders, wenn die Elektrodenkabel lang sind. Zudem besteht ein Instrumentenverstärker aus recht vielen Bauteilen. Wir haben uns deshalb für einen Kompromiss zwischen Einfachheit und Empfindlichkeit entschieden und einen Differenzverstärker (IC2a) anstelle eines Instrumentenverstärkers verwendet.

Am Ausgang des Differenzverstärkers gelangt das Signal zu einem Bandpassfilter zweiter Ordnung (IC2b und IC2c), das unerwünschte Störsignale entfernt. Es besitzt eine Gesamtverstärkung von etwa 2,4 und lässt Signale zwischen 20 Hz und 500 Hz passieren. Alles außerhalb dieses Bereichs wird mit 12 dB/Oktave abgeschwächt.

Das gefilterte Signal wird von IC2d weiter verstärkt und anschließend von der Diode D1 gleichgerichtet. Kondensator C9 extrahiert die Hüllkurve des EMG-Signals. R15 ist ein Entladewiderstand für C9, der ansonsten „überlaufen" würde. Abschließend wird die Hüllkurve durch IC3a verstärkt und das Signal von C10 (mit Entladewiderstand R17) geglättet. Am Ausgang, der am Analog-Digital-Wandler (ADC) des Arduino-Mikrocontrollers angeschlossen ist, ist im Signalweg eine 2,8-V-Zenerdiode (D2) eingeschleift. Diese Diode verhindert eine falsche Triggerung der Software aufgrund von statischen Änderungen des EMG-Signals.

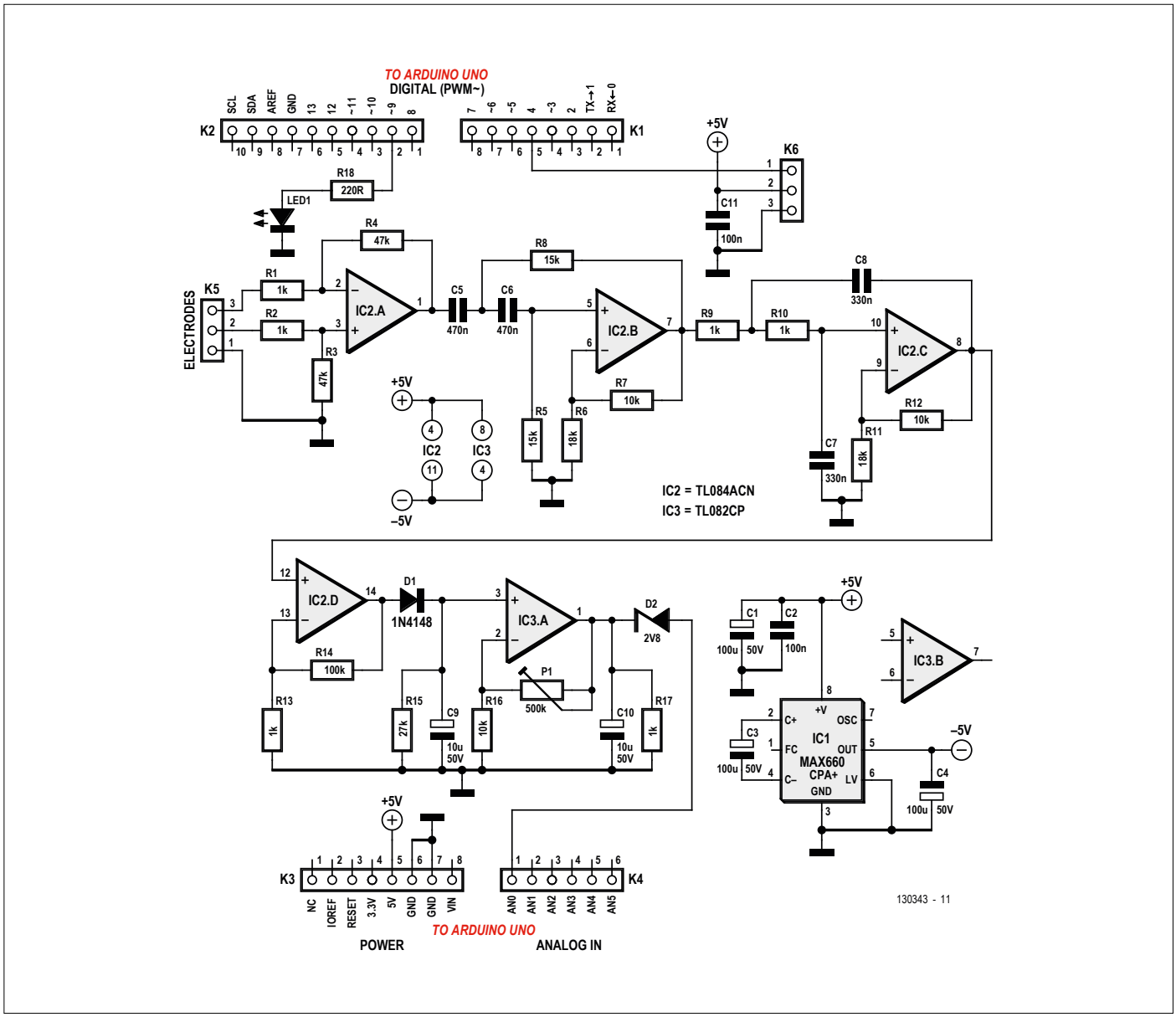


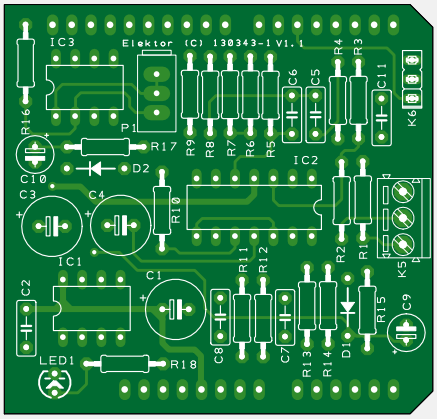

Bild 2. Ein Differenzverstärker, ein Bandpassfilter und ein Hüllkurvenfolger bereiten das EMG-Signal für die Digitalisierung vor.

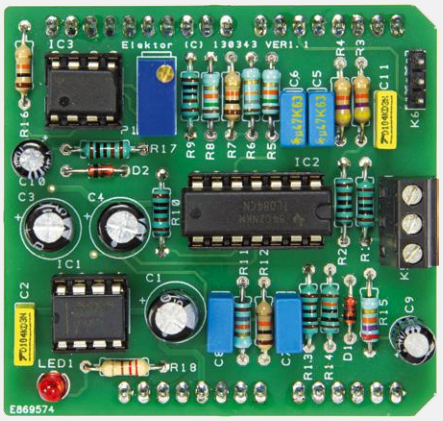
## STÜCKLISTE

**Widerstände:**
alle 250 mW, 5 %
R1,R2,R9,R10,R13,R17 = 1 k
R3,R4 = 47 k
R5,R8 = 15 k
R6,R11 = 18 k
R7,R12,R16 = 10 k
R14 = 100 k
R15 = 27 k
R18 = 220 Ω
P1 = Trimmpoti 500 k, 24-Gang

**Kondensatoren:**
C1,C3,C4 = 100 µ, 50 V, Raster 2 mm
C2,C11 = 100 n, Raster 0,2“
C5,C6 = 470 n, 50 V, Raster 0,2“
C7,C8 = 330 n, 50 V, Raster 0,2“
C9,C10 = 10 µ, 50 V, Raster 2 mm

**Halbleiter:**
D1 = 1N4148




D2 = 1N5224B-TP
IC1 = MAX660CPA
IC2 = TL084ACN
IC3 = TL082CP
LED = rot, 3 mm

**Außerdem:**
K1,K3 = 1x8-polige Stiftleiste, Raster 0,1“
K2 = 1x10-polige Stiftleiste, Raster 0,1“
K4 = 1x6-polige Stiftleiste, Raster 0,1“
K5 = 3-polige Platinenanschlussklemme, Raster 3,5 mm
K4 = 1x3-polige Stiftleiste, Raster 0,1“
Platine 130343-1

Die Verstärkung von IC3a ist variabel und kann mit dem Mehrgangtrimmer P1 präzise eingestellt werden. Die LED zeigt an, wenn P1 auf das Leerlaufsignal (das heißt, bei entspanntem Muskel) richtig abgestimmt ist.
Die Platine für die Schaltung ist als 5-V-Arduino-Shield ausgelegt. Der Spannungswandler MAX660 (IC1) invertiert die Versorgungsspannung, um eine symmetrische ±5-V-Betriebsspannung für das Shield zu erhalten.

### Software

Die Software wurde in der Arduino-IDE für den Arduino Uno geschrieben. Das analoge Eingangssignal am ADC-Eingang Pin A0 wird in einen numerischen Wert gewandelt und mit der Map-Funktion des Arduino in den Bereich 0...180 skaliert, korrespondierend mit den Winkelgraden, um die sich der Servo drehen soll. Die skalierten ADC-Werte werden an den Servomotor (beziehungsweise die Servo-Bibliothek des Arduinos) weitergegeben. Der Servomotor selbst ist an Pin 9 des Arduino Uno angeschlossen.

Übrigens, Servomotoren haben meist drei Anschlussdrähte: Plus (+), Minus (-) und das Steuersignal. Normalerweise, aber nicht immer, ist das mittlere Kabel Plus. Vergewissern Sie sich also vor Inbetriebnahme, dass die Kabel korrekt angeschlossen sind.

### Ein Wort zur Sicherheit

Die Schaltung entspricht natürlich keineswegs den Vorschriften für Medizinelektronik. So verlockend es auch sein mag, den Arduino mit dem USB-Port eines netzbetriebenen Computers zu verbinden, so sehr sollte man deshalb davon Abstand nehmen. Bei einem Laptop ist es unproblematisch, aber wenn der Computer an das Stromnetz angeschlossen ist, kann es theoretisch eine Verbindung zwischen den Elektroden und der Steckdose geben. Das ist gefährlich bis tödlich für die Person, die die Elektroden trägt. Daher **sollte man die Schaltung nur mit einer 9-V-Batterie oder einer 5-V-USB-Powerbank versorgen. Vermeiden Sie sämtliche Verbindungen zum Stromnetz!**

### Das Versuchskaninchen ist dran!

Wenn Sie ein bereitwilliges Versuchskaninchen gefunden haben, können Sie die Schaltung testen. Stecken Sie das Shield auf den Arduino und verbinden Sie die Elektroden mit dem Shield. Bringen Sie die Elektroden an einem Arm- oder Beinmuskel des Probanden an. Die betreffenden Hautstellen sollten zuvor mit Alkohol gereinigt/entfettet werden, damit die Elektroden gut haften. Die dritte Elektrode sollte mit der Shield-Masse verbunden sein. Wenn man diese Elektrode vergisst oder falsch anbringt, werden die Verstärker schnell gesättigt und Muskelkontraktionen nicht erkannt.
Es ist auch wichtig, dass der Proband isolierende Schuhe trägt oder auf einer Gummimatte steht, um von der Schutzerde (PE) isoliert zu sein. Ansonsten könnten Störungen das Referenzsignal beeinträchtigen und merkwürdige Resultate erzeugen.
Der Proband ist nun ganz entspannt (vor allem der muscle under test, MUT), während Sie das Trimmpoti P1 so einstellen, dass die LED aufleuchtet. Die LED kann ein wenig blinken, aber Dauerlicht ist besser.
Wenn der Proband den Muskel zusammenzieht, sollte der Servo drehen, während die LED verlischt. Nach dem Entspannen des Muskels sollte der Servomotor zurückkehren und die LED wieder einschalten. ⏮

(130343)

### Weblink

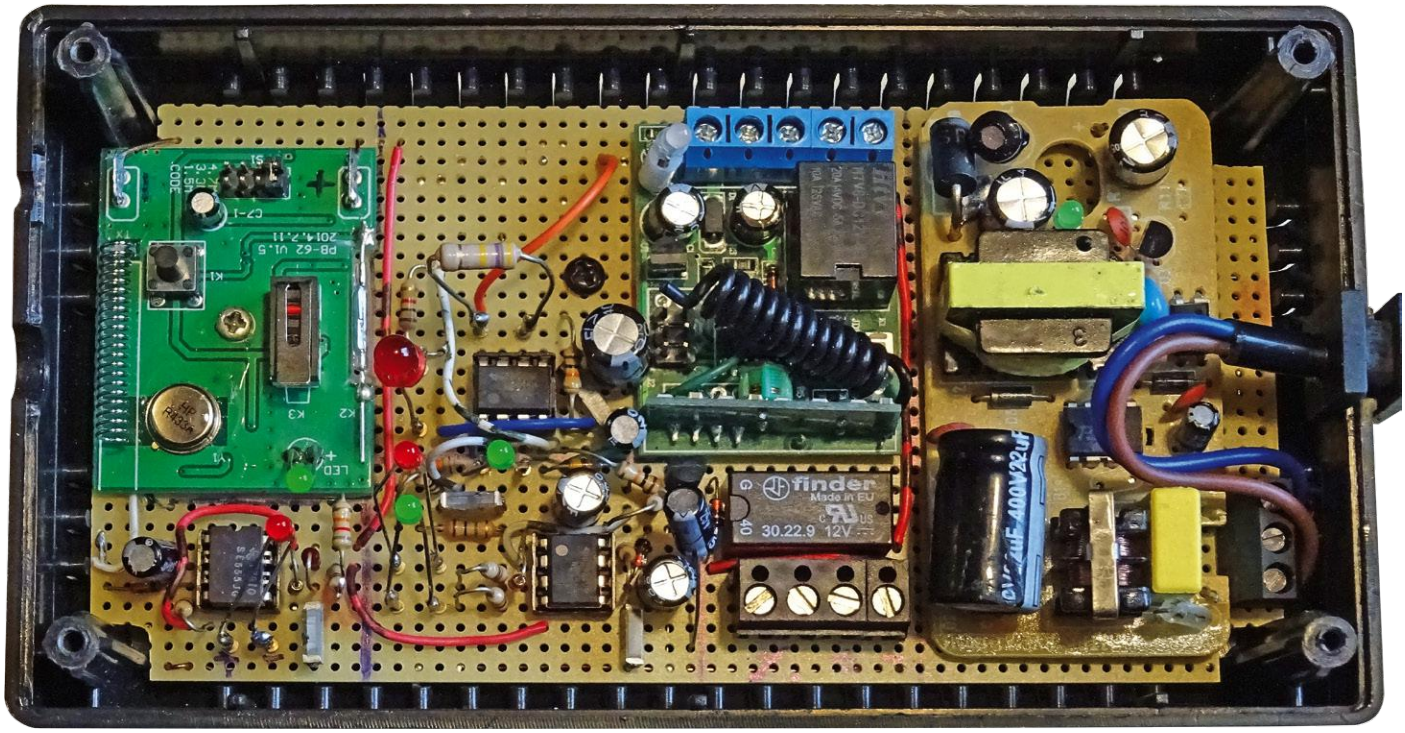
[1] www.elektormagazine.com/130343

## EINKAUFSZETTEL

→ 130343-1
unbestückte Platine

# Jammer-Alarm

## Meta-Alarmanlage schützt vor Störsendern




Die weit verbreiteten, günstigen und leicht zu installierenden Funk-Alarmanlagen haben einen großen Nachteil: Ein Bösewicht kann sie problemlos mit einem Störsender schachmatt setzen. Der Jammer-Alarm ist eine Alarmanlage der Metaebene: Er warnt, wenn ein Störsender die Funk-Alarmanlage stört!

Von **Walter Meyer**

Die Schwachstelle von Funk–Alarmanlagen dürfte nach den vielen Veröffentlichungen in den Medien sicher bekannt sein. Mit einem Jammer (Störsender) oder einem 433-MHz-Handfunksender lässt sich der Funkverkehr der Alarmanlage nachhaltig stören. Es reicht, einen Träger zu setzen und an der Anlage kommt kein Signal der ohnehin recht schwachen Funkalarmmelder mehr an. Welch ein Jammer!
Solche Störsender werden in der Internet-Werbung angepriesen als Mittel, den Mobilfunkverkehr, GPS, WLAN und ähnliche Funkmedien zum Schutz der Privatsphäre zu blockieren. Um es klar zu sagen: **Solche Störsender dürfen in der Europäischen Union (und der USA und in Japan) von Privatpersonen oder Unternehmen weder beworben, vertrieben, gekauft und schon gar nicht angewendet werden [1].** Es geht dabei nämlich keinesfalls um den Schutz der Privatsphäre. Stattdessen dienen die mittlerweile preiswerten und kleinen Geräte in der Regel kriminellen Handlungen: Das Ortungssystem eines gestohlenen Fahrzeugs wird außer Betrieb gesetzt oder ein Funk-Alarmsystem eines Wohnhauses wird neutralisiert. Außerdem können durch einen Jammer leicht lebensrettende Maßnahmen wie Notrufe an Feuerwehr oder Polizei gestört oder verhindert werden! Insofern scheinen auch die gravierenden Sanktionen gerechtfertigt: Wer beim Betrieb eines Störsenders erwischt wird, kann mit Strafen im fünfstelligen Eurobereich rechnen!

### Funkstrecke in the box

Die Schaltung des Jammer-Alarms wartet geradezu darauf, gestört zu werden, um dann über eine Sirene und eine Drahtleitung zur Alarmanlage Alarm zu schlagen. Sie besteht, wie man in **Bild 1** sehen kann, aus fünf, und wenn man das Netzteil mitzählt, aus sechs Funktionsblöcken. Drei davon sind nicht ohne Grund grün eingefärbt, es handelt sich nämlich um fertige Module, die aus alten Geräten ausgeschlachtet und/oder so billig im Internet erworben werden können, dass sich ein Selbstbau bei weitem nicht lohnt. Doch nun der Reihe nach.

Oben links befindet sich ein 555-Timer, der als astabiler Multivibrator beschaltet ist und alle 13 Sekunden einen kurzen negativen Impuls zu einem 433,92-MHz-Sender schickt. Dabei handelt es sich um einen gewöhnlichen Funk-Tür/Fensterkontakt. Im Schaltbild ist dieser als erstes vorgefertigtes „grünes“ Element und auf dem Titelbild als grüne Platine zu erkennen. Das Signal des Timers wird hier an der spannungsabgewandten Seite des NO-Reedkontakts angeschlossen. Je nach Schaltung des Tür/Fensterkontakts kann das auch anders aussehen: Der negative Impuls muss das Gleiche bewirken wie das Öffnen eines alarmgesicherten Fensters. Zwei LEDs signalisieren, dass der Sender wie gewünscht Impulse auf einem 433-MHz-Träger über die Antenne (im Titelbild ganz außen) verschickt. Die grüne LED des Türkontakts leuchtet und verlischt kurz während des Impulses, währenddessen die rote LED aufblitzt. Der kleine Taster auf dem Sender ermöglicht, wenn vorhanden, einen manuellen Funktionstest.

Das Funksignal wird von der Antenne eines kleinen, lernfähigen Universal-Funkempfängers für 433 MHz mit einem Kanal aufgefangen. Solche Empfänger gibt es im Internet (zum Beispiel bei ebay) zuhauf für wenig Geld (10...15 €, manchmal auch zu noch niedrigeren „Kampfpreisen“) zu kaufen. Die Empfänger sind dafür eingerichtet, ASK-kodierte Signale zu empfangen, ihren Code zu akzeptieren und entsprechend ein Relais zu schalten. Bei dem von mir verwendeten Modell OC-433L1 lassen sich verschiedene Schaltmodi des Relais (Taster, Wechsler, Schalter, Timer mit verschiedenen Zeiten) einstellen. Da das Modul nur den empfangenen Impuls (beim OC-433L1 angezeigt von einer weißen LED) wieder ausgeben soll, reicht die Funktion „Taster“ aus. Das Relais auf der Platine wird überhaupt nicht benötigt und sollte zwecks Geräuschminimierung ausgebaut oder deaktiviert werden.

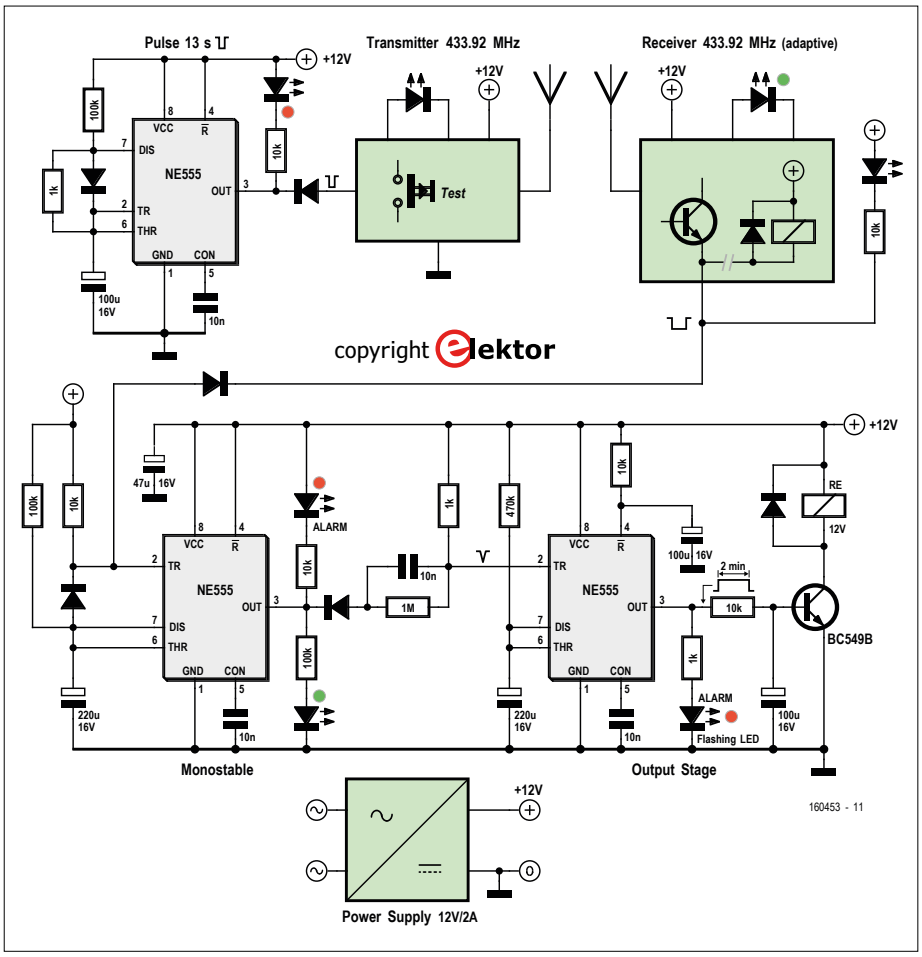



Bild 1. Die Funktionsblöcke des Jammer-Alarms.

## Impulsformung

Nun geht es wieder ans Selbstgemachte! Ein weiteres Timer-IC NE555 ist als retriggerbarer Multivibrator geschaltet. Sein Ausgang ist im Normalfall High (die grüne LED an seinem Ausgang leuchtet) und kippt nach etwa 25 Sekunden auf Low (die rote Alarm-LED leuchtet). Daran wird er aber im Normalfall, wenn die Funkstrecke nicht gestört ist, durch die regelmäßigen, im 13-s-Rythmus erscheinenden negativen Impulse gehindert. Nur wenn die Funkstrecke unterbrochen ist und die Impulse ausbleiben, kann der Multivibrator auf Low kippen.

Der letzte Funktionsblock ist die Ausgangsstufe; sie ist ebenfalls um einen als monostabiler Multivibrator geschalteten 555-Timer aufgebaut. Wenn er getriggert wird, kippt sein Ausgang für etwa zwei Minuten auf High. Über das Relais kann eine Sirene aktiviert und/oder ein Alarm an der Alarmanlage ausgelöst werden. Als Alarm-LED habe ich eine Blink-LED verwendet, deren Datenblatt eine Vorwärtsspannung von 3,5...5 V und einen Strom von 8...20 mA angibt. Der Vorwiderstand sollte also in der Größenordnung von 1 kΩ liegen. Bei anderen LEDs kann der Wert anders sein. Konsultieren Sie das Datenblatt der Blink-LED!

Über ein Detail gibt es noch Interessantes zu berichten, nämlich den 10-nF-Kondensator zwischen den beiden Timern. Es handelt sich um ein Differenzierer. Kippt der erste monostabile Multivibrator im Alarmfall auf Low, wird nur die negative Flanke als kurzer negativer Impuls an die Ausgangsstufe weitergegeben. Damit wird verhindert, dass die Ausgangsstufe dauerhaft (und nicht nur für zwei Minuten) Alarm auslöst.

Über die letzte „fertige“ Baugruppe in der Schaltung gibt es nicht viel zu erzählen. Es handelt sich um das Innere eines ganz gewöhnlichen Steckernetzteils, das +12 V Gleichspannung bei einem Strom von 2 A liefern kann.

## Im Test

Ich habe das Gerät bei einem Funkamateur mit zwei 433-MHz-Handgeräten und verschiedenen Sendeleistungen ausgiebig getestet. Der Jammer-Alarm verhält sich genau so, wie ich es geplant (und gehofft) hatte. Wird die Funkstrecke gestört, löst das Gerät nach Ausbleiben des Funksignals zuverlässig Alarm aus.

(160453)

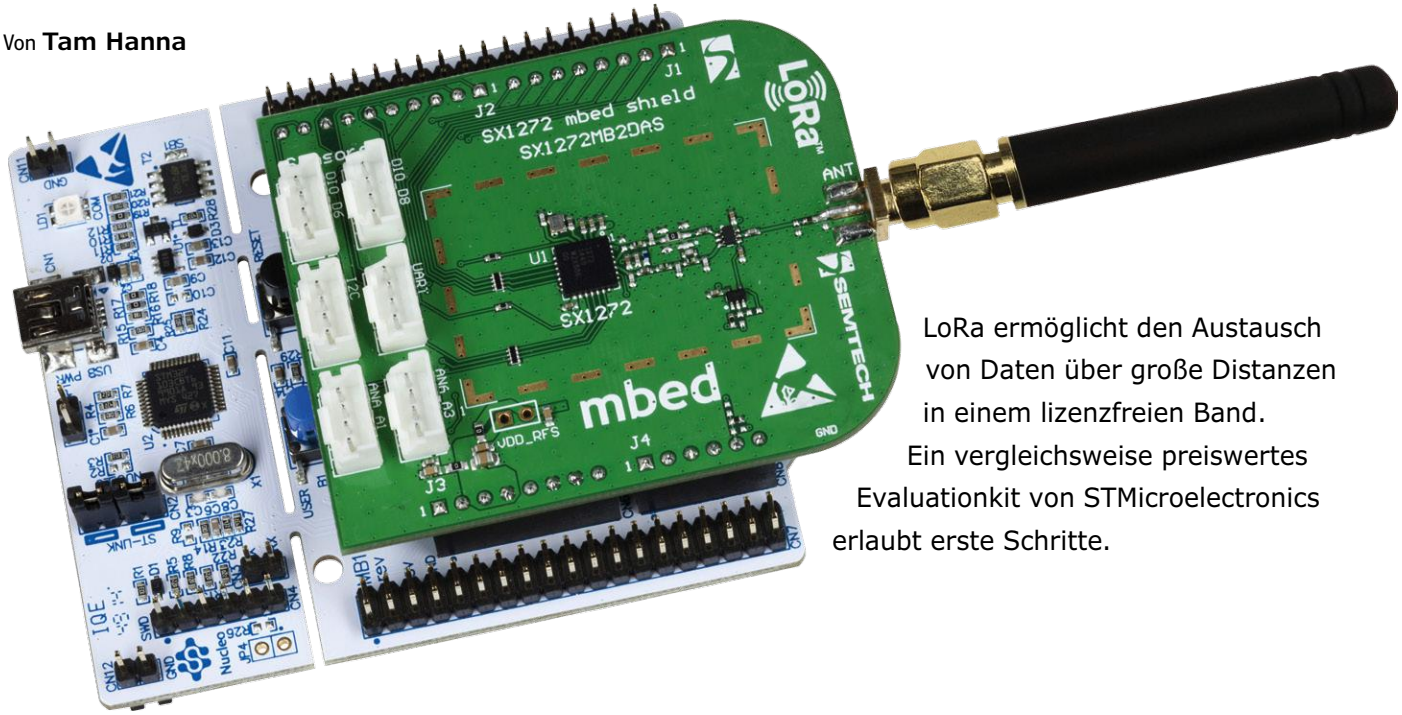
## Weblinks


[1] www.bundesnetzagentur.de/cln_1411/SharedDocs/Pressemitteilungen/DE/2016/160624_Stoersender.html

# Erste Schritte mit LoRa

## Kurzstreckenfunk mit zwei Eval-Boards

Von **Tam Hanna**




LoRa ermöglicht den Austausch von Daten über große Distanzen in einem lizenzfreien Band. Ein vergleichsweise preiswertes Evaluationkit von STMicroelectronics erlaubt erste Schritte.

Die Arbeit mit LoRa ist insofern schwierig, als man normalerweise ein (teures) Gateway braucht. Doch erste Schritte sind auch ohne Anschluss ans Internet möglich. Das von STM angebotene SX1272-Shield passt auf diverse Nucleo-Boards, und erweitert diese um einen vollwertigen LoRa-Stack. Darüber hinaus wird auch ein Kit aus Shield, Antenne und Controllerboard angeboten (siehe **Kasten** „Was brauchen Sie?“). Dieser Artikel zeigt, wie Sie zwei derartige Platinen zum Datenaustausch einsetzen können.
Besonders wertvoll ist, dass sich STM nicht auf das Liefern von Hardware beschränkt: Wer die Boards erwirbt, darf auf eine umfangreiche Sammlung an Referenzanwendungen zurückgreifen. Wir nutzen eine davon als Basis für ein kleines Projekt, das durch Toggeln von Portpins das Ermitteln der Signallaufzeit erlaubt.

### Erste Schritte

Wer das Kit erwirbt, muss nur noch die mitgelieferte Antenne mit dem Modul verbinden und ein Mini-USB-Kabel bereitstellen, mit dem eine Verbindung zu einem PC aufgebaut wird. In den folgenden Schritten nutzen wir als Entwicklungsumgebung das kostenlose OpenSTM32 (System Workbench) und die CUBE-Libraries zur Abstraktion von der Hardware. Freunde von Keil können natürlich auch auf deren IDE setzen.
STM ist nicht für seine besonders übersichtliche Webseite bekannt. Der erste Schritt zum funktionsfähigen Funkmodul ist das Herunterladen des LoRA-Stacks, der unter [1] zum Download bereitsteht (siehe **Bild 1**).
Extrahieren Sie das rund 100 MB große Archiv in das Verzeichnis *C:\Users\TAMHAN\STM32Cube* - statt *TAMHAN* setzen Sie Ihren Benutzernamen ein. STM spendiert im Rahmen des MDK drei verschiedene Beispielapplikationen. Neben ClassA, das zum Realisieren eines (mit einem Gateway zu verbindenden) Clients vorgesehen ist, gibt es mit AT_Slave auch eine Implementierung eines Modems.

Für uns ist an dieser Stelle allerdings nur PingPong interessant: Dieses Programm realisiert eine Verbindung zwischen zwei gleichartigen LoRA-Modulen, und kommt dabei ohne (teures) Gateway aus.
Da der STM-eigene LoRA-Stack eine Vielzahl verschiedener Controllerboards, eine Vielzahl verschiedener Module und drei verschiedenen IDEs unterstützt, ist die Verzeichnisstruktur relativ komplex. Das für uns interessante Projekt sitzt im Verzeichnis *Projects\Multi\Applications\LoRa\PingPong\SW4STM32\STM32L073RZ-Nucleo\sx1272mb2das*. Importieren Sie es im ersten Schritt wie gewohnt in den Workspace, um sich an der in **Bild 2** gezeigten Projektstruktur zu erfreuen.
Senden Sie das Programm sodann mit *Run -> Ac6 STM32 Debugging* auf das erste der beiden Controllerboards. Die IDE wird den Code erfolgreich ausliefern und ausführen. Beim Debugging kommt man in OpenSTM32 normalerweise nur bis zum obligatorischen Breakpoint im Einsprungpunkt von `main()`. Danach kommt es aufgrund eines unter [2] dokumentierten Fehlers zu permanenten Resets, die sich in der Konsole mit Ausgaben nach folgendem Schema äußern:

```
Polling target stm32l0.cpu failed, trying to
   reexamine

Examination failed, GDB will be halted. Polling again
   in 6300ms
```

Zum Debugging ist die Verwendung von Keil oder IAR angeraten – da wir im Moment aber ohne Debugger auskommen, wollen wir uns nun gleich dem eigentlichen Verbindungsaufbau zuwenden. Hierzu stecken wir das erste der beiden Controllerboards ab, um das zweite ebenfalls unter Nutzung von Run mit der PingPong-Applikation zu versorgen.

## Daten abernten

Auf ein mit dem PC verbundenes Nucleo-Board kann man unter Windows über einen seriellen Port zugreifen. Zum Entgegennehmen der angelieferten Informationen setzen wir - wie von STM empfohlen - auf das in Japan entwickelte TeraTerm, das unter [3] zum Download bereitsteht.
Installieren Sie das Programm wie gewohnt auf ihrem PC, um es daraufhin aus dem Startmenü heraus anzuwerfen. Verbinden Sie das erste Controllerboard mit einer Energieversorgung: Ein Handyladegerät mit einem absteckbaren Ladekabel leistet hier großartige Dienste. Wer sich „mobilisieren" will, setzt stattdessen auf eine Powerbank – wenn sich diese aufgrund des geringen Stromverbrauchs des STM32 abschalten sollte, kann man einen Widerstand parallelschalten.
Das Board Nummer Zwei wird nun mit dem PC verbunden. Starten Sie TeraTerm, und klicken Sie auf Datei ➞ Neue Verbindung. Aktivieren Sie den Radiobutton „Seriell" und wählen Sie den Port aus, der zum STM-Board passt. TeraTerm verbindet sich, gibt anfangs aber nur Abfall aus. Zur Lösung des Problems klicken Sie auf Einstellungen ➞ Serieller Port, und stellen die in **Bild 3** gezeigten Parameter ein.
Beachten Sie, dass STM für die verschiedenen Applikationen unterschiedliche Baudraten vorgesehen hat: Die mit einem Gateway arbeitenden Programme gehen wesentlich schneller zu Werke. Nach erfolgreicher Einstellung der Baudrate erscheinen sinnvolle Werte – die Ausgabe sieht je nach PC etwas anders aus, orientiert sich aber an folgendem Schema:

```
OnRxDone
        RssiValue=-11 dBm, SnrValue=29
                                        rxDone
...PONG
       OnTxDone
               txDone
```

## Was passiert hier?

Nachdem wir wissen, dass das Programm funktioniert, können wir uns dem Code zuwenden. Die Methode `main` beginnt mit der Initialisierung diverser Hardwarekomponenten, um sich sodann dem Radio zuzuwenden:

```
int main( void )
{

  . . .

  RadioEvents.TxDone = OnTxDone;
  RadioEvents.RxDone = OnRxDone;
  RadioEvents.TxTimeout = OnTxTimeout;
  RadioEvents.RxTimeout = OnRxTimeout;
  RadioEvents.RxError = OnRxError;
  Radio.Init( &RadioEvents );
  Radio.SetChannel( RF_FREQUENCY );
```

**Was brauchen Sie?**

Das von STMicroelectronics als P-NUCLEO-LRWAN1 bezeichnete Evaluationkit besteht aus einer Antenne, einem LoRa-Modul und einem STM32-Mikrocontrollerboard [6]. Zum Nachvollziehen der hier besprochenen Anweisungen sind derer zwei erforderlich – bei Mouser kostete das Kit zum Zeitpunkt der Drucklegung rund 65 €.

Die Radio-API von STM arbeitet komplett asynchron.
Die `RadioEvents`-Struktur liefert eine Gruppe von Funktionen an, die vom Stack aufgerufen werden, wenn ein bestimmtes Ereignis auftritt.

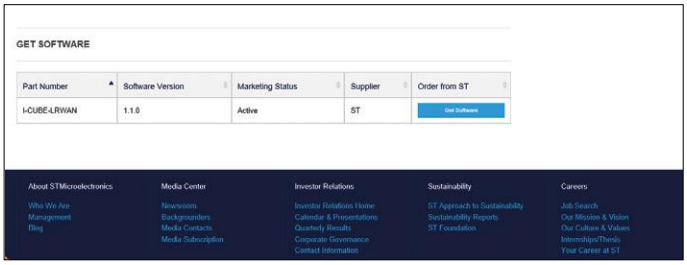



Bild 1. Der Download-Link steht unten auf der Website.

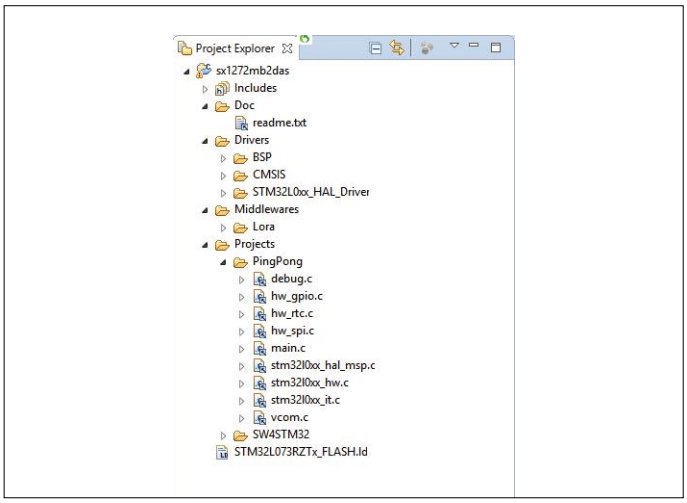



Bild 2. Der LoRa-Stack steht zum Einsatz bereit.

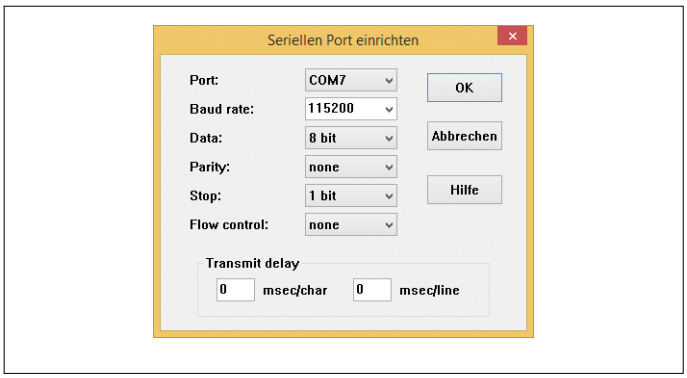



Bild 3. Die Baudrate von 115200 bps ist vergleichsweise gemütlich.

Bevor wir uns der Endlosschleife mit der eigentlichen Programmlogik zuwenden, ein kurzer Blick auf eine dieser Handlerfunktionen. `OnRxDone` wird immer dann aufgerufen, wenn Informationen erfolgreich vom Funkmodul entgegengenommen wurden.

```
void OnRxDone( uint8_t *payload, uint16_t size,
   int16_t rssi, int8_t snr )
{
    Radio.Sleep( );
    BufferSize = size;

    memcpy( Buffer, payload, BufferSize );
    RssiValue = rssi;
    SnrValue = snr;
    State = RX;

  . . .

    PRINTF(„OnRxDone\n“);
    PRINTF(„RssiValue=%d dBm, SnrValue=%d\n“, rssi,
   snr);

}
```

Die anderen Methoden verhalten sich im Großen und Ganzen ähnlich. Der einzige Unterschied ist, dass sie keine Payload entgegennehmen und kompakter ausfallen. Damit können wir zum Hauptprogramm zurückkehren; die Endlosschleife ist in Form eines Zustandsautomaten aufgebaut. Für Quereinsteiger: Es handelt sich dabei um ein Programm, das permanent zwischen mehreren Zuständen wechselt. Die Realisierung derartiger - im Embedded-Bereich sehr häufig verwendeter - Programmstrukturen erfolgt normalerweise über eine *Enum*, die die Zustände auflistet. Im Falle unseres Programms sieht diese folgendermaßen aus:

```
typedef enum
{
    LOWPOWER,
    RX,
    RX_TIMEOUT,
    RX_ERROR,
    TX,
    TX_TIMEOUT,
}States_t;
```

Beginnen wir mit dem Zustand eines Timeouts nach dem Senden von Informationen: In diesem Fall wird das Radio dazu angewiesen, Daten entgegenzunehmen. Die RX-Methode wird mit einem Wert dotiert, der die Zeit angibt, nach der ein weiterer Timeout ausgelöst werden sollte:

```
case TX_TIMEOUT:

    Radio.Rx( RX_TIMEOUT_VALUE );
    State = LOWPOWER;
    break;
```

Das PingPong-Programm arbeitet mit einem selbst konfigurierenden Verbund aus Meistern und Sklaven. Im Fall eines Problems beim Empfangen von Informationen weist der Master das Funkmodul an, ein weiteres Ping-Paket zu senden. Ein Sklave kehrt einfach in den Empfangszustand zurück:

```
case RX_ERROR:

  if( isMaster == true ) {

    Buffer[0] = ‚P‘;
    Buffer[1] = ‚I‘;
    Buffer[2] = ‚N‘;
    Buffer[3] = ‚G‘;

    for( i = 4; i < BufferSize; i++ ) {
      Buffer[i] = i - 4;
    }

    Radio.Send( Buffer, BufferSize );
  }

  else {
    Radio.Rx( RX_TIMEOUT_VALUE );
  }

  State = LOWPOWER;
  break;
```

Damit können wir uns dem wichtigsten Problem zuwenden: dem RX-Zustand. Beginnen wir mit der Betrachtung des Zustands als Master – aus Platzgründen ist der hier abgedruckte Code stark gekürzt:

```
case RX:
  if( isMaster == true ){
    if( BufferSize > 0 ){
      if( strncmp( ( const char* )Buffer, ( const
   char* )PongMsg, 4 ) == 0 )
      {

        . . .

        Buffer[0] = ‚P‘;
        Buffer[1] = ‚I‘;
        Buffer[2] = ‚N‘;
        Buffer[3] = ‚G‘;

        for( i = 4; i < BufferSize; i++ ){
          Buffer[i] = i - 4;
            }

        PRINTF(„...PING\n“);
        DelayMs( 1 );
        Radio.Send( Buffer, BufferSize );
        }
```

Der erste Fall betrifft einen Master, der eine Pong-Message von einem Sklaven erhält. Seine Reaktion ist das Senden einer weiteren Ping-Nachricht, um die Kommunikation aufrechtzuerhalten.

Schwieriger wird die Lage beim Eintreffen eines Ping-Pakets: dieser Zustand weist darauf hin, dass schon ein anderer Master

existiert. Die Reaktion der Beispielimplementierung beschränkt sich in diesem Fall darauf, sofort in den Sklavenzustand zu verfallen:

```
    else if( strncmp( ( const char* )Buffer, ( const
   char* )PingMsg, 4 ) == 0 ){
      isMaster = false;
      Radio.Rx( RX_TIMEOUT_VALUE );
    }

    else {
      isMaster = true;
      Radio.Rx( RX_TIMEOUT_VALUE );
    }
  }
}
```

Damit können wir uns der Rolle des Sklaven zuwenden. Der Empfang einer Ping-Nachricht führt logischerweise zur Emission eines Pongs:

```
    else{
      if( BufferSize > 0 ){
        if( strncmp( ( const char* )Buffer, ( const
   char* )PingMsg, 4 ) == 0 ){

          Buffer[0] = ‚P‘;
          Buffer[1] = ‚O‘;
          Buffer[2] = ‚N‘;
          Buffer[3] = ‚G‘;
          for( i = 4; i < BufferSize; i++ ){
            Buffer[i] = i - 4;
          }

          DelayMs( 1 );
          Radio.Send( Buffer, BufferSize );
          PRINTF(„...PONG\n“);

        }
```

Interessanter ist die Frage, was beim Empfang einer andersartigen Nachricht passiert. Da die Implementierung auf einen Meister und einen Sklaven beschränkt ist, interpretiert STM dies als Aufforderung zum Wechsel in den Mastermodus:

```
        else {
          isMaster = true;
          Radio.Rx( RX_TIMEOUT_VALUE );
        }
      }
  }

  State = LOWPOWER;
  break;
```

## Mach‘ etwas mehr

Wir wollen uns nun der Frage zuwenden, wie lang der Austausch von Datenpaketen über eine optimale Verbindung dauert. Hierzu ist ein Digitalspeicheroszilloskop ideal. Zur Messung wandeln wir das eben besprochene Programm so ab, dass beim Senden des Masters einer der Ausgänge des Controllerboards getog-

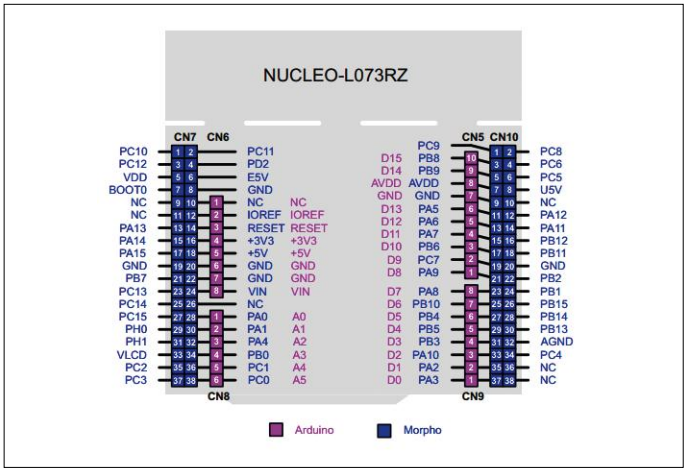



Bild 4. Einige der Pins sind zwischen Morpho- und Arduino-Connector geteilt (Grafik: STMicroelectronics).

gelt wird. Wenn der Slave ein Datenpaket empfängt, lassen wir einen anderen Ausgangspin umschalten.
Das Controllerboard im Kit hört auf den Namen NUCLEO-L073RZ: Im Dokument UM1724 [4] bietet der Hersteller Informationen zum Pinout der GPIO-Ports an. Einen Ausschnitt davon zeigt **Bild 4**.
Wir verwenden wieder dasselbe Programm auf beiden Boards. Beim Programmstart initialisieren wir einfach beide Portpins:

```
GPIO_InitTypeDef GPIO_InitStructure;

int main( void ) {

. . .

  HAL_Init( );

  __GPIOC_CLK_ENABLE();
  GPIO_InitStructure.Pin    = GPIO_PIN_5 | GPIO_PIN_8;
  GPIO_InitStructure.Mode   = GPIO_MODE_OUTPUT_PP;
  GPIO_InitStructure.Pull   = GPIO_NOPULL ;
  GPIO_InitStructure.Speed  = GPIO_SPEED_HIGH;

  HAL_GPIO_Init(GPIOA, &GPIO_InitStructure);
```

Interessant ist hier eigentlich nur ein kleiner Trick: `GPIO_InitStructure`-Instanzen lassen sich in aktuellen Versionen von CUBE zur gleichzeitigen Initialisierung mehrerer Pins einspannen.
Der nächste Akt ist das Toggeln der Ausgänge (siehe dazu den **Kasten** „Wieso Toggeln?“). Als Erstes schalten wir Pin 5 des Masters, wann immer eine Nachricht gesendet wird:

```
case RX:

    if( isMaster == true ){
    if( BufferSize > 0 ) {
        if( strncmp( ( const char* )Buffer, ( const
   char* )PongMsg, 4 ) == 0 ) {

        . . .
```

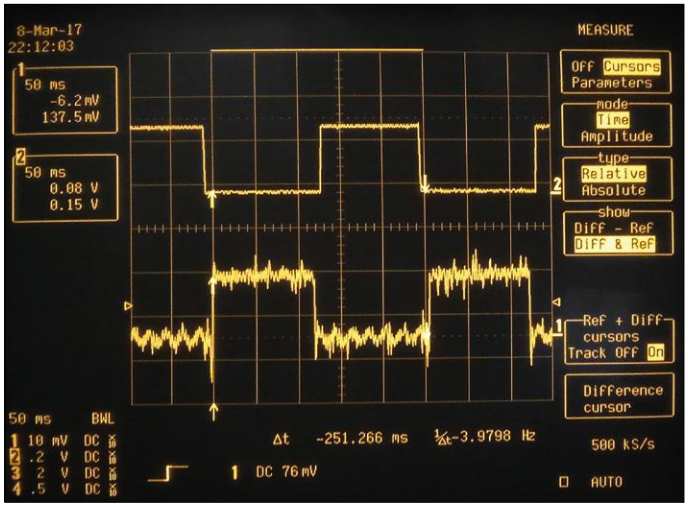



Bild 5. Ein Sende-Empfangszyklus nimmt unter optimalen Bedingungen rund 250 ms in Anspruch.

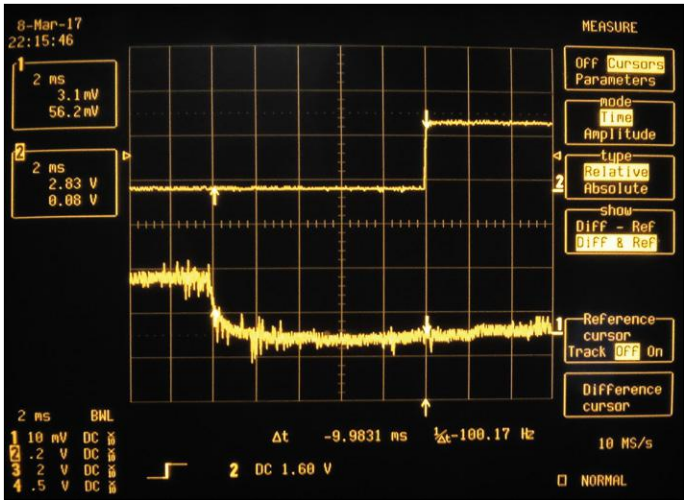



Bild 6. Zwischen den beiden Toggles vergehen – unter optimalen Bedingungen – rund 10 ms.

```
        HAL_GPIO_TogglePin(GPIOC, GPIO_PIN_5);
        Radio.Send( Buffer, BufferSize );
```

Schritt zwei ist das Toggeln von Pin 8, wenn der Slave Informationen entgegennimmt:

```
else if( strncmp( ( const char* )Buffer, ( const
   char* )PingMsg, 4 ) == 0 ) {

    isMaster = false;
    Radio.Rx( RX_TIMEOUT_VALUE );
    HAL_GPIO_TogglePin(GPIOC, GPIO_PIN_8);
}
```

Nachdem wir das geänderte Programm (Download unter [5]) auf beiden Boards gestartet haben, sind wir zur Aberntung der Informationen bereit. Pin 8 des Boards, das die Sklavenrolle einnimmt, wird an CH1 des Oszilloskops angeschlossen, während Pin 5 des anderen Boards an CH2 kommt.
Achten Sie darauf, dass – so gut wie alle - Oszilloskope für sämtliche Kanäle denselben Groundpegel verwenden, weshalb die Stromversorgung der beiden Controllerboards identische Potentiale aufweisen muss. Die sodann erscheinenden Ergebnisse unserer Laufzeitmessung sind in **Bild 5** und **Bild 6** gezeigt.

## Eine Frage der Reichweite

Auch wenn es verlockend ist: Wir wollen hier keinen Blick auf die Reichweite werfen. Die von STM bereitgestellten Transceiver würden keine Rückschlüsse auf das Verhalten von LoRa als Ganzes erlauben – die in der Presse gemeldeten Rekordergebnisse erfolgen unter Nutzung wesentlich teurerer Antennen. Angegeben werden kann allerdings, dass sich LoRa auch mit den von STM gestellten und nicht sonderlich qualitativen Antennen dem (durchaus leistungsstarken) WLAN-Router des Autors massiv überlegen zeigt. In der stahlbetonverseuchten Wohnung fand sich kein einziger Ort, an dem der Empfang gestört war.

## Fazit

STMs Hardware ist mit rund 130 € im Doppelpack noch bezahlbar: Wer erste Schritte in die Welt von LoRa gehen möchte, wird den Kauf der Entwicklungsplatinen mit Sicherheit nicht bereuen. Ob des geringen Preises fällt es vergleichsweise leicht, über die Schwächen der kostenlosen Entwicklungsumgebung hinwegzusehen.
Das hier gezeigte Beispielprogramm und die STM32-Boards bieten zudem diverse Möglichkeiten der Erweiterung. Selbstverständlich könnte man zum Beispiel Sensoren auslesen und die Messwerte übertragen - der Autor freut sich über Lesermails (tamhan@tamoggemon.com) mit Ideen. ◀

(160271)

### Wieso toggeln?

Als Elektroniker denkt man bei einer Laufzeitmessung zuerst an das Emittieren von Pulsen. Dies ist insofern nicht sinnvoll, als die Erfassung kurzer Impulse bei langen Laufzeiten die Nutzung einer Peak-Detect-Funktion voraussetzt.
Der Autor bietet unter [7] ein englischsprachiges Video an, das die Thematik näher erklärt.

## Weblinks

[1] www.st.com/content/st_com/en/products/embedded-software/mcus-embedded-software/stm32-embedded-software/stm32cube-expansion-software/i-cube-lrwan.html

[2] http://nuttx.org/doku.php?id=wiki:howtos:jtag-debugging

[3] https://ttssh2.osdn.jp/index.html.en

[4] www.st.com/content/ccc/resource/technical/document/user_manual/98/2e/fa/4b/e0/82/43/b7/DM00105823.pdf/files/DM00105823.pdf/jcr:content/translations/en.DM00105823.pdf

[5] www.elektormagazine.de/160271

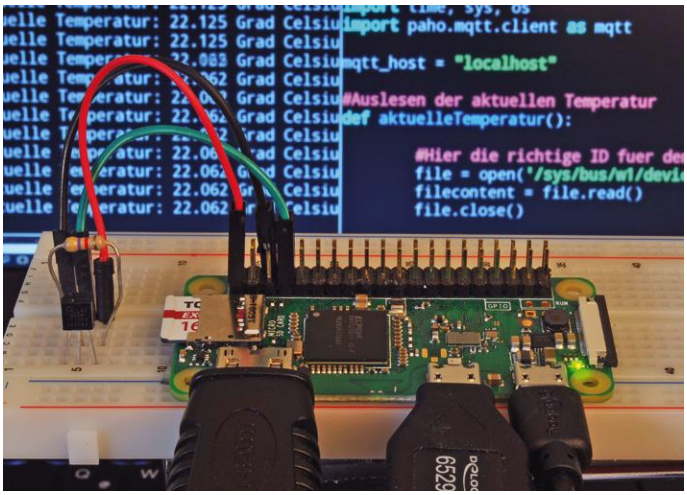
[6] www.st.com/en/evaluation-tools/p-nucleo-lrwan1.html

[7] www.youtube.com/watch?v=EirGe5LzSkc

# Klein, aber oho: Raspberry Pi Zero W

## Erste Schritte mit der Mini-Himbeere

Die Mini-Version des Raspberry Pi mit dem Namenszusatz Zero gibt es bereits seit Ende 2015. Im Februar ist nun eine um WLAN und Bluetooth ergänzte Variante Zero W („Wireless“) erschienen. In diesem Artikel schauen wir uns an, mit welchen Schritten man die neue Platine an den Start bringt, was sie leistet und welche Vorteile die Funktechnologie in Verbindung mit dem winzigen Single-Board-Computer ermöglicht. In einem Praxisbeispiel schließen wir einen Temperatursensor an, und verschicken die Daten per MQTT-Protokoll über WLAN.




Von **Markus Ulsaß** (D)

Mit einem Preis von 10 Dollar plus Steuern ist der neue Raspberry Pi Zero W zwar doppelt so teuer wie sein Vorgänger, ergänzt diesen aber um die für viele Projekte unverzichtbaren Funkdienste WLAN und Bluetooth. Damit werden günstige Roboter-, Smart-Home- und Internet-of-Things-Projekte möglich, die von der kleinen Bauform und einem recht niedrigen Stromverbrauch des Zero W profitieren. Das einkernige Broadcom-SoC BCM2835 wird mit 1 GHz getaktet (512 MB RAM). Damit ist der Zero W zwar nicht so schnell wie sein großer Bruder Raspberry Pi 3, aber für die meisten Projekte trotzdem mehr als ausreichend.
In Deutschland ist pi3g [1] der offizielle Distributor für den Zero W. Als Alternative bietet sich der britische Shop Pimoroni [2] an, wo trotz Versand kaum höhere Kosten bei einem Kauf entstehen und auch die Lieferung recht zügig ist.
Der Zero W ist mit Mini-HDMI- und Micro-USB-On-The-Go-Ports ausgestattet; die Einzelplatine wird ohne bestückten 40-Pin-Header (HAT-kompatibel) geliefert. Daher bietet es sich an, sofern man die entsprechenden Adapter nicht schon in der Bastelkiste liegen hat, gleich das entsprechende Zubehör mitzubestellen, welches als Kit von den Shops angeboten wird.

### Installation des Betriebssystems und erste Schritte

Hat man einen Zero W ergattert und alle benötigten Adapter sowie eine Stromversorgung vorliegen (benötigtes Zubehör siehe Kasten „Materialliste“), muss man sich zunächst das Betriebssystem auf eine mindestens 8 GB große Micro-SD-Karte installieren.

Dazu lädt man sich unter [3] das aktuelle Image „Raspbian Jessie with Pixel“ herunter. Um das Image auf eine SD-Karte aufzuspielen, benötigt man ein Tool wie Etcher, welches unter [4] heruntergeladen werden kann (**Bild 1**). Der Vorteil dieses Programms ist übrigens, dass es für alle Betriebssystem-Plattformen erhältlich ist und das heruntergeladene Image nicht entpackt werden muss. Hat man das Programm installiert und gestartet, wählt man zunächst das passende Image aus und überträgt es dann auf die ausgewählte SD-Karte.
Danach steckt man die SD-Karte in den Zero W, schließt per Adapter den Monitor (über HDMI) sowie ein Keyboard und eine Maus (über USB) an und steckt zuletzt die Stromversorgung an. Achtung! Der Anschluss dafür ist der rechte der beiden Micro-USB-Buchsen, wenn man das Board waagerecht vor sich liegen hat und sich die SD-Karte links befindet (**Bild 2**).
Jetzt sollte der Raspberry Pi Zero W in den Pixel-Desktop des Raspbian-Betriebssystems starten. Damit hat man sein lauffähiges Linux-Betriebssystem auf dem Kleinstrechner fertig

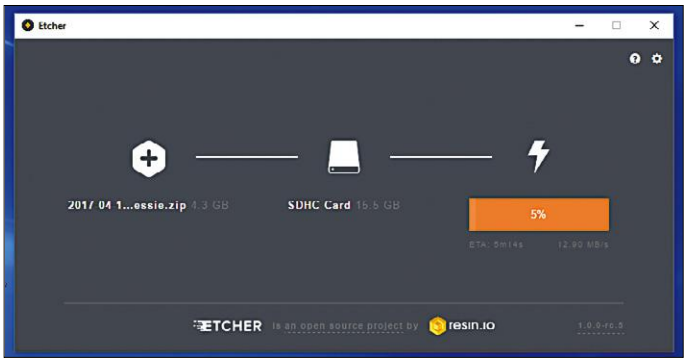



Bild 1. Etcher ist ein SD-Karten-Tool zum Aufspielen des Raspbian-Image.

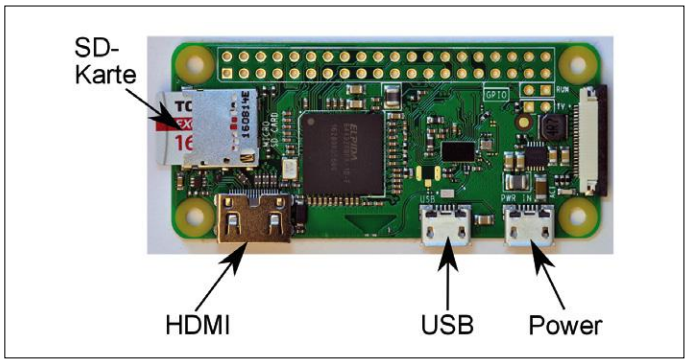



Bild 2. Anschlüsse des Raspberry Pi Zero W.

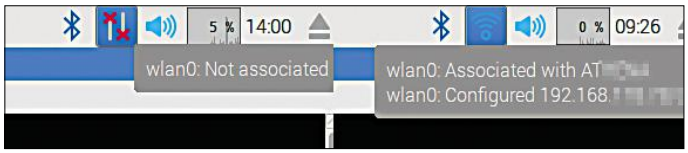



Bild 3. Vorher (links)/ nachher (rechts). Nach der Konfiguration sollte sich der Zero W mit dem WLAN verbinden. Zusatzinformationen bekommt man oft, indem man die Maus über die Symbole bewegt.

aufgesetzt.

## Feintuning

Wenn man nun ein Terminal öffnet (*Strg+T* oder das Monitor-Icon oben in der Menüleiste anklicken), wird man feststellen, dass keine deutsche Sprachumgebung und kein deutsches Tastatur-Layout eingestellt sind. Das können wir ändern, indem wir *sudo raspi-config* im Terminal eintippen (den Bindestrich erzeugt man mit dem ß auf der deutschen Tastatur). Danach öffnet sich das *Raspberry Pi Software Configuration Tool* und dort wählen wir mit dem Cursor nach unten *4 Localisation Options* aus und drücken auf *Enter*. Wir wählen in *I1 Change Locale de_DE.UTF-8 UTF-8* mit der *Leertaste* aus und gehen per *Tab* auf *<Ok>*. Im nachfolgenden Fenster setzen wir *de_DE.UTF-8* als *Default locale for the system evironment* und bestätigen wieder mit *<Ok>*. Wir starten den Pi Zero W mit *sudo reboot* neu.

Das System hat nun die deutsche Sprachumgebung als Vorgabe, allerdings ist das Keyboard-Layout noch nicht korrekt. Wir rufen wieder *sudo raspi-config* auf, wählen *4 Localisation Options* aus und wählen diesmal *I3 Change Keyboard Layout* aus. Dort setzen wir *Generische PC-Tastatur mit 105 Tasten (Intl)*, im nächsten Fenster *Andere* und dann *Deutsch* und bestätigen mit *<Ok>*. Wir gehen mit dem Cursor auf *Deutsch – Deutsch (T3)* und bestätigen wieder per *Enter*-Taste. Die nachfolgenden Fenster bestätigen wir ebenfalls mit *<Ok>* bzw. *<Nein>* und kommen am Ende wieder in das Hauptmenü. Wenn wir dieses mit *<Finish>* verlassen und nun auf der Konsole zum Beispiel ein *z* eingeben, sollte das deutsche Tastaturlayout eingestellt sein. Da der Zero W das Standard-Passwort *raspberry* für den Benutzer *pi* als Vorgabe hat, sollten wir dies unbedingt ändern. Das geschieht mit *sudo raspi-config* und dem Eintrag *1 Change User Password.* Wir vergeben ein neues Passwort und notieren es uns. Damit der Zero W später im Netzwerk auch per Namen eindeutig erkannt werden kann, ändern wir diesen (Standard ist *raspberrypi*). Wir editieren dafür in zwei Textdateien den entsprechenden Eintrag, und zwar mit dem Texteditor nano. Zuerst geben wir *sudo nano /etc/hostname* ein und ersetzen in der Textdatei den Eintrag *raspberrypi* durch *zerow*. Wir speichern mit *Strg+O, Enter* und beenden mit *Strg+X*. Danach öffnen wir mit *sudo nano /etc/hosts* die zweite Datei und suchen den letzten Eintrag mit *127.0.0.1* auf. Auch hier ersetzen wir wieder *raspberrypi* durch *zerow*. Danach speichern wir ebenfalls wieder mit *Strg+O, Enter* und beenden mit *Strg+X*.

## Verbindung nach außen

Damit sich unser Zero W mit dem lokalen Netzwerk per WLAN verbinden kann, müssen wir nun noch die Daten für den WLAN-Router (SSID, Passwort) eingeben. Das können wir durch Anklicken des Symbols mit den zwei senkrechten Linien und roten Kreuzen rechts oben in der Menüleiste machen (**Bild 3**). Wir wählen den richtigen Access-Point aus, geben das Passwort ein und der Raspberry Pi Zero W sollte sich mit dem WLAN verbinden.

Eine sinnvolle Maßnahme mit einem frischen Image ist, die

**Listing 1. LED-Blink-Python-Skript.**

```
#!/usr/bin/python
#LED_Blink.py
import RPi.GPIO as GPIO #Einbinden der GPIO-Bibliothek
import time #Bibliothek, die fuer den Sleep benoetigt wird

LED = 14
GPIO.setmode(GPIO.BCM) #BCM-GPIO Bezeichnungen benutzen
GPIO.setwarnings(False) #Warnungen ausschalten
GPIO.setup(LED, GPIO.OUT) #Pin der LED als Ausgang nutzen
PAUSEON = 1.0 #An-Zeit
PAUSEOFF = 1.0 #Aus-Zeit

while True:
    GPIO.output(LED, GPIO.HIGH) #LED anschalten
    time.sleep(PAUSEON) #An-Zeit
    GPIO.output(LED, GPIO.LOW) #LED ausschalten
    time.sleep(PAUSEOFF) #Aus-Zeit
```

### MATERIALLISTE

- Raspberry Pi Zero W
- Micro-SD-Karte (min. 8 GB)
- HDMI-auf-miniHDMI-Adapter
- Micro-USB-OTG-Adapter (z.B. Delock 65296)
- Micro-USB-Netzteil (min. 1 A)
- 2x20-Pin-Stiftleiste (2,54 mm Rastermaß)
- Steckbrett
- Steckbrücken
- LED (rot)
- Widerstand 470 Ω
- Widerstand 4k7
- Temperatursensor DS18B20

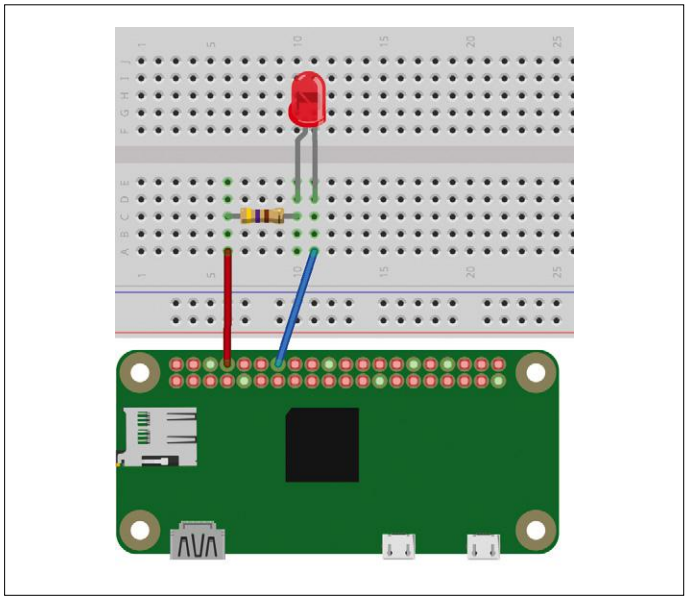


Bild 4. Der Fritzing-Sketch für den Anschluss der LED an den Zero.

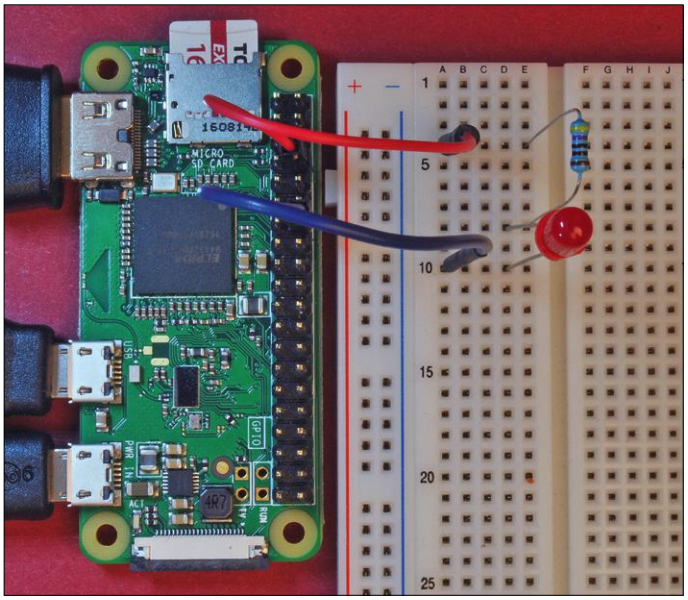


Bild 5. Die Verbindungen des Zero W mit LED und Widerstand auf dem Steckbrett.

Paketquellen und das System auf den neuesten Stand zu bringen. Das erledigen wir mit *sudo apt update && sudo apt upgrade* wieder in einem Terminal. Wir bestätigen die Frage nach dem Fortfahren mit *J* und warten ein paar Minuten, bis alle Downloads durchgeführt sind.

## „Hallo Welt“ per LED

Nachdem wir die grundlegenden Vorbereitungen abgeschlossen haben, machen wir uns daran, erste einfache Hardware anzuschließen. Dazu müssen wir zunächst die Stromversorgung trennen den 40-Pin-Header an den Zero W anlöten, um per Kabelbrücken und Steckbrett eine LED anzusteuern.
Vor dem Anschließen des Widerstands und der LED fahren wir den Zero W komplett herunter, um Kurzschlüsse oder andere negative Effekte beim Hinzufügen von Hardware direkt an die GPIOs zu vermeiden. Dies erledigen wir mit *sudo shutdown –h now* auf der Konsole oder indem wir per Klick auf das Himbeer-Symbol in der Menüleiste im Drop-Down Menü den Punkt *Shutdown* und im nachfolgenden Fenster *Herunterfahren* anwählen.

Für unser „Hallo Welt“ per LED benutzen wir zur Ansteuerung den GPIO14, einen 470-Ohm-Widerstand sowie eine rote LED. Das Pinout des Zero W findet man zum Beispiel unter [5]. Wir verbinden den GPIO14 mit dem Widerstand, diesen mit der Anode der LED und deren Kathode mit einem der Ground-Pins am Pi Zero W. **Bild 4** zeigt den Fritzing-Sketch, **Bild 5** demonstriert den Aufbau in natura.

Als Programm zum Blinken der LED nutzen wir ein kleines Python-Skript (LED_Blink.py). Es ist in **Listing 1** abgedruckt, kann aber auch von der Elektor-Website unter [6] heruntergeladen werden.
Wir können den Code wieder mit dem Editor *nano* oder auch mit dem etwas komfortableren *Geany* (*Menü -> Entwicklung -> Geany*), der standardmäßig im Raspbian-Betriebssystem enthalten ist, erzeugen. Zu beachten ist hierbei, dass keine Umlaute benutzt werden dürfen (gilt vor allem für die Kommentare) und die Einrückungen (Indentation) bei Python wie im Listing per Leertaste erzeugt werden müssen, da es sonst zu Fehlermeldungen (IndentationError) kommt. Nach dem Abspeichern des Programms rufen wir es auf der Kommandozeile im Terminal mit *python LED_Blink.py* auf. Die LED sollte nun im Sekundenrhythmus blinken.
Beenden können wir das Programm mit *Strg+C*.

## Temperatur messen mit dem DS18B20

Ein gängiger und einfach anzuschließender digitaler Temperatur-Sensor ist der DS18B20 von Maxim Integrated. Zunächst fahren wir das System herunter, um die Schaltung auf dem Steckbrett sicher aufzubauen und mit dem Zero W zu verbinden. Der linke Pin (Typenbezeichnung vorne) des DS18B20 wird mit der 3,3-V-Versorgungsspannung des Zero W verbunden, der rechte mit einem Groundpin. Vom mittleren (Daten-)Pin wird eine Steckbrücke zu GPIO4 gelegt. Schließlich kommt noch ein

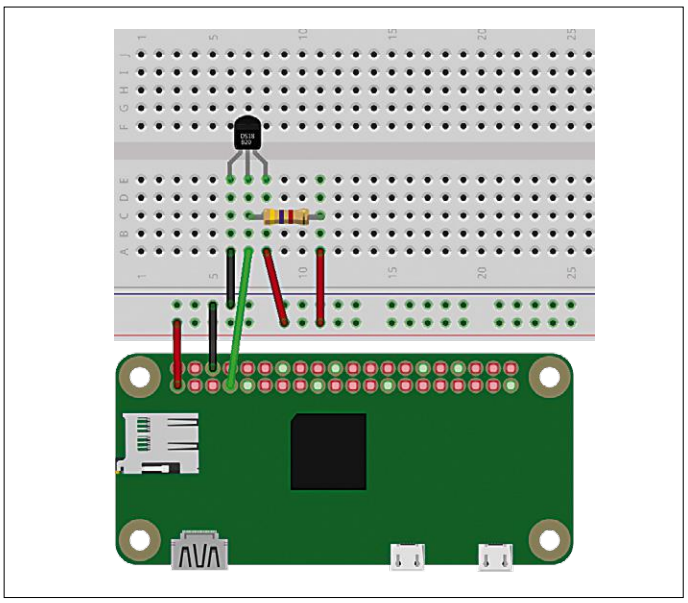


Bild 6. Der Schaltungsaufbau mit dem Temperatursensor DS18B20.

```
pi@zerow:~ $ cd /sys/bus/w1/devices
pi@zerow:/sys/bus/w1/devices $ ls
28-00000362eca7  w1_bus_master1
pi@zerow:/sys/bus/w1/devices $ cd 28-00000362eca7
pi@zerow:/sys/bus/w1/devices/28-00000362eca7 $ cat w1_slave
60 01 4b 46 7f ff 10 10 b5 : crc=b5 YES
60 01 4b 46 7f ff 10 10 b5 t=22000
pi@zerow:/sys/bus/w1/devices/28-00000362eca7 $
```

Bild 7. Testen des angeschlossenen Sensors.

Pull-Up-Widerstand zwischen Datenpin und Versorgungsspannung (siehe das Fritzing-Schema in **Bild 6**).
Nach dem Anschluss und einer Kontrolle, ob auch alles richtig verdrahtet ist, starten wir unseren Zero W wieder. Bevor wir jedoch den One-Wire-Temperatur-Sensor nutzen können, muss dieser zunächst in das System integriert werden. Das machen wir, indem wir mit *sudo raspi-config* unter *P7 1-Wire* die Frage nach *Would you like the one-wire interface to be enabled?* mit *Ja* beantworten. Anschließend beenden wir das Konfigurationsprogramm und müssen das System rebooten.
Jetzt testen wir, ob der Sensor richtig in das System eingebunden und angeschlossen ist. Dazu wechseln wir per Terminal in das Verzeichnis *cd /sys/bus/w1/devices* und lassen uns mit *ls* den Inhalt auflisten. Dabei sollte eine alphanumerische Kombination erscheinen, die unseren Sensor repräsentiert.
Wir wechseln mit *cd <Sensor-ID>*, also in unserem Fall mit *cd 28-00000362eca7* in das entsprechende Unterverzeichnis und lassen uns mit *cat w1_slave* die aktuellen Daten des Sensors anzeigen, die in zwei Zeilen ausgegeben werden (**Bild 7**). t=22000 bedeutet in diesem Fall eine Temperatur von 22 Grad Celsius.
Damit ist der Sensor richtig konfiguriert und betriebsbereit.

## Temperatursensor an den Rest der Welt per MQTT

Sensorwerte, die wir an unserem Raspberry Pi Zero W erfassen, möchten wir unter Umständen an andere Empfänger weiterleiten. Dazu nutzen wir das schlanke MQTT-Protokoll – ein Nachrichtenprotokoll für die Maschine-zu-Maschine-Kommunikation. MQTT verwendet ein so genanntes Publish-Subscribe-System. Dabei können beispielsweise Sensoren Daten unter bestimmten Topics veröffentlichen (Publish), Clients können sich auf diese Topics abonnieren (Subscribe) und die Daten empfangen. Die Topics erinnern mit den enthaltenen Schrägstrichen an Dateipfade (ein Beispiel wäre zum Beispiel „Wohnung/Küche/Temperatur"), es sind sogar Wildcards möglich. Statt Files werden hier aber eben verschiedene Sensoren bzw. auch Aktoren spezifiziert.
Unser Temperatursensor könnte die gemessenen Temperaturwerte unter dem Topic „/haus/aussen/temperatur/sensor1" versenden, auf den sich ein Client – z.B. ein Hausautomationssystem wie openHAB, fhem oder Node-Red abonniert hat. Für MQTT wird ein Vermittler – im MQTT-Jargon *Broker* genannt - benötigt, der den Datenfluss zwischen Sender und Empfänger organisiert. Ein populärer Broker ist Mosquitto, den wir mit *sudo apt install mosquitto mosquitto-clients* auf unserem Zero W installieren.
Zur Kontrolle, ob alles richtig installiert ist, können wir sowohl das Versenden als auch das Empfangen von Daten auf dem gleichen Gerät testen. Dazu müssen wir zunächst den Broker mit *sudo systemctl start mosquitto* auf der Konsole starten. Damit er auch bei einem Neustart automatisch aktiviert wird, geben wir noch *sudo systemctl enable mosquitto* ein.
Wir simulieren auf der Konsole mit *mosquitto_sub -h localhost -t /sensor1* (wir sparen uns für die folgenden Beispiele etwas Schreibarbeit und verkürzen das oben vorgestellte Topic) zunächst einen Nachrichtenabonnenten, der später Daten empfangen (mosquitto_sub) soll. Es passiert zunächst nichts, da das System ja auf Daten wartet. Wir öffnen ein zweites Terminal-Fenster und geben *mosquitto_pub -h localhost -t /sensor1 -m „22"* ein (**Bild 8**). Damit tun wir so, als ob wir einen Temperaturwert versenden (mosquitto_pub), bzw. um im MQTT-Vokabular zu bleiben, veröffentlichen (Publish). Im ersten Fens-

```
pi@zerow:~ $ mosquitto_sub -h localhost -t /sensor1
22
```

```
pi@zerow:~ $ mosquitto_pub -h localhost -t /sensor1 -m "22"
pi@zerow:~ $
```

Bild 8. Testen der MQTT-Funktion.

```
pi@zerow:~ $ python DS18B20_MQTT.py
Connected with MQTT-Broker (IP): localhost
Current temperature: 23.187 degrees Celsius
Current temperature: 23.187 degrees Celsius
Current temperature: 23.187 degrees Celsius
Current temperature: 23.187 degrees Celsius
Current temperature: 23.187 degrees Celsius
Current temperature: 23.187 degrees Celsius
Current temperature: 23.187 degrees Celsius
Current temperature: 23.187 degrees Celsius
```

```
pi@zerow:~ $ mosquitto_sub -h localhost -t /sensor1
23.187
23.187
23.187
23.187
23.187
23.187
23.187
23.187
```

Bild 9. Test des MQTT-Datenversands unseres Temperatursensors: Links die lokale Ausgabe, rechts der Empfang der Werte über MQTT.

ter sollte daraufhin eine *22* erscheinen. Unser System für das Veröffentlichen unseres Temperatursensor-Wertes steht nun und in einem letzten Schritt automatisieren wir es noch.

### Versenden der Temperatur per MQTT

Wir benutzen ein Python-Skript, um die Daten automatisiert per MQTT zu versenden. Dafür benötigen wir eine Bibliothek, die wir mit *sudo pip install paho-mqtt* installieren.

Unser Skript (**Listing 2**) importiert zunächst die benötigten Module und definiert die IP des Rechners, auf dem der Mosquitto-Broker läuft („localhost" für den lokalen Broker, IP-Adresse für einen entfernten MQTT-Broker).
Dann wird der aktuelle Temperatur-Wert in der Funktion `currentTemperature()` ausgelesen. Für die im Beispielcode angegebene Sensor-ID muss natürlich die des verwendeten Sensors eingesetzt werden. Die ausgelesenen Daten werden in einen passenden Wert umgewandelt und als Rückgabewert der Funktion ausgegeben. Danach wird eine Verbindung zum lokalen bzw. entfernten MQTT-Broker (mqtt_host) aufgebaut. In der while-Schleife wird dann alle zwei Sekunden der Temperaturwert (`temperature`) ausgelesen, auf der Kommandozeile ausgegeben und unter dem Topic /sensor1 an den MQTT-Broker gesendet.

**Listing 2. Python-Skript zum Versenden der Temperatur per MQTT.**

```
#!/usr/bin/python
#DS18B20_MQTT.py
#Import der benoetigten Module
import time, sys, os
import paho.mqtt.client as mqtt

mqtt_host = "localhost"

#Auslesen der aktuellen Temperatur
def currentTemperature():

    #Hier die richtige ID fuer den Sensor eingeben
    file = open('/sys/bus/w1/devices/28-00000362eca7/w1_slave')
    filecontent = file.read()
    file.close()

    #Temperaturwert umwandeln
    temperaturestring = filecontent.split("\n")[1].split(" ")[9]
    temperaturevalue = float(temperaturestring[2:]) / 1000
    return(temperaturevalue)

def on_connect(client, userdata, flags, rc):

    print("Connected with MQTT-Broker (IP): " + mqtt_host)

client = mqtt.Client()
client.on_connect = on_connect

client.connect(mqtt_host, 1883, 60)

client.loop_start()

while True:
    time.sleep(2)

    temperature = currentTemperature()
    print("Current temperature: " + str(temperature) + " degrees
Celsius")
    client.publish("/sensor1", temperature)
```

Nachdem wir das Programm mit einem Editor erstellt bzw. heruntergeladen haben, öffnen wir ein Terminal und starten es mit *python DS18B20_MQTT.py*. Auf der Kommandozeile sollten eine Verbindung mit dem MQTT-Broker auf dem lokalen Zero W gemeldet und danach die aktuellen Temperaturwerte angezeigt werden (**Bild 9**).

Nun erzeugen wir ein zweites Terminal und „subscriben" uns wieder per MQTT-Client auf das Topic „/sensor1" auf dem lokalen Rechner: *mosquitto_sub -h localhost -t /sensor1*.

### Das alles und noch viel mehr

Mit dem Zero W ist der Raspberry-Pi-Foundation pünktlich zum fünften Geburtstag des Ur-Raspberry Pi wirklich ein tolles Board gelungen. Wer hätte vor fünf Jahren gedacht, dass ein solcher Linux-Kleinstrechner in so einer winzigen Ausführung zu dem geringen Preis möglich ist?
Der USB-Port, der beim Vorgänger noch durch Tastatur oder WLAN-Modul blockiert war, ist nun für weitere Funktionalität verfügbar. Auch der Stromverbrauch ist sehr gering und erreicht schon fast die Bereiche für selbstversorgte Internet-of-Things- bzw. Smart-Home-Geräte. Einen Vergleich des Stromverbrauchs verschiedener Raspberry Pis findet man unter [7].
Mit all diesen Eckdaten ist es nicht verwunderlich, dass das Board nur in geringen Stückzahlen zu bekommen ist. Das war schon beim ersten Raspberry Pi so und daher kann man nur hoffen, dass sich das Angebot an Zero-W-Boards bald weiter verbessert. ◀

(160451)

### Weblinks


[1] www.buyzero.de

[2] https://shop.pimoroni.com/

[3] www.raspberrypi.org/downloads/raspbian/

[4] https://etcher.io/

[5] https://pinout.xyz/

[6] www.elektormagazine.de/160451

[7] https://blog.adafruit.com/2017/ 03/03/how-much-power-does-pi-zero-w-use-piday-raspberrypi-raspberry_pi/

eLABS PROJECT

# LED-Treiber für 12 V

## Bringt Licht in dunkle Kajüten

Wenn die Dämmerung in der Bootskajüte oder im Wohnmobil Einzug hält, sind Niedervolt-Glühlampen auch heute noch die verbreiteten Lichtspender. Nachteilig ist an dieser Art der Beleuchtung, dass Glühlampen schlechte Wirkungsgrade haben, so dass der Bordakku beizeiten erschöpft ist. LED-Leuchten für 12 V sind zwar erheblich sparsamer, sie sind aber auch deutlich teurer. Der Autor hat deshalb einen Treiber entworfen, der den Betrieb einer preiswerten 230-V-LED-Leuchte an 12 V möglich macht.

Von **Louter van der Kolk** (NL)

Der schwedische Möbelriese Ikea hat eine kleine LED-Klemmleuchte im Sortiment, die den Namen „Jansjö" trägt und rund 10 € kostet. Diese Leuchte wäre für das Boot oder das Wohnmobil ideal, wenn sie nicht für den Betrieb an 230 V (mit einem Adapter) ausgelegt wäre. Um die Leuchte am Bordnetz mit 12 V betreiben zu können, bedarf es einer klügeren Lösung.

### Eigenschaften

- Betrieb am 12-V-Bordnetz
- Geeignet für Leuchten mit High-Efficiency-LEDs
- Einstellbarer Ausgangsstrom bis 350 mA
- Dimm-Modi mit Lötbrücken wählbar

### Suboptimal

Vor noch nicht allzu langer Zeit wurden sogenannte Wechselrichter eingesetzt, um aus der Gleichspannung 12 V die Wechselspannung 230 V zu erzeugen. Damit war der Einsatz von Haushaltsglühlampen oder Leuchtstoffröhren in Wohnmobilen oder auf Booten möglich. Die Wechselspannung 230 V lässt sich natürlich in eine Gleichspannung umformen, an der LED-Leuchten betrieben werden können. Doch halt – zuerst mit einem Wechselrichter auf hohe Spannung umformen, und dann mit einem Netzadapter auf niedrige Spannung? Diese Methode könnte in ein Buch mit dem Titel „Elektronik für Unbedarfte" gehören, aber nicht in ein Fachblatt wie Elektor...

### Deutlich besser

Die zeitgemäße Lösung besteht darin, einen integrierten LED-Treiberbaustein zu verwenden, der aus der Gleichspannung 12 V des Bordnetzes ohne Umweg den Strom für die LED-Leuchte bereitstellt. Als LED-Leuchte ist „Jansjö" von Ikea besonders geeignet, denn der Adapter für 230 V Wechselspannung und die Leuchte sind über eine Steckverbindung miteinander verbunden. Ohne Adapter führt die Anschlussleitung direkt zur LED. Der LED-Strom im normalerweise vorgesehenen Betrieb lässt sich mit jedem Multimeter schnell messen, er beträgt bei „Jansjö" ungefähr 165 mA. Ein LED-Treiber muss daher als Stromquelle arbeiten, er muss aus der Gleichspannung 12 V den Strom 165 mA generieren.

### Treiber-Schaltung

Natürlich ist es möglich und nebenbei auch lehrreich, einen LED-Treiber vollständig diskret aufzubauen, mit einzelnen Transistoren, Kondensatoren, Widerstän-

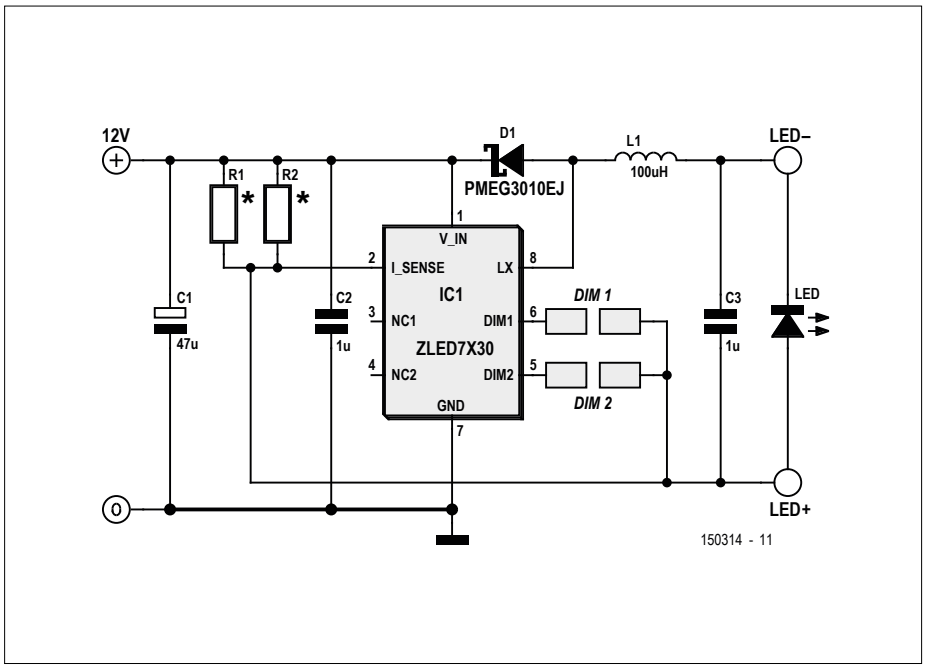



Bild 1. Die Schaltung besteht aus einem IC und wenigen externen Komponenten.

den und Induktivitäten. Im Internet sind zu diesem Thema zahlreiche Vorschläge zu finden. Doch ehrlich: Das ist nicht mehr „State of the Art“, insbesondere nicht, wenn die aufgebaute Schaltung klein, leicht und kompakt ausfallen soll. Wir haben deshalb nach einem integrierten Treiber-Baustein Ausschau gehalten, der die nötigen Funktionen auf einem Chip vereint. Gefunden haben wir die Produktreihe ZLED7030 von Integrated Device Technology (IDT). Die genaue Typenbezeichnung hängt vom Ausgangsstrom ab, **Tabelle 1** gibt Auskunft. Dieses unscheinbare IC, das nur wenige externe Bauelemente benötigt, bietet einige interessante Möglichkeiten. Dazu gehört beispielsweise die Dimm-Funktion, sie wird mit dem Schalter des angeschossenen Verbrauchers gesteuert. Der Ausgangsstrom lässt sich mit einem einzelnen Widerstand präzise einstellen. Leider wird dieses IC nur in SMD-Bauform gefertigt, es lässt sich jedoch noch von Hand löten.

Da der Strombedarf der Ikea-LED-Leuchte „Jansjö“ nur 165 mA beträgt, kommt hier der Typ ZLED7730 zum Einsatz. Wenn eine andere LED-Leuchte einen höheren Strom verlangt, kann aus Tabelle 1 der dazu passende Typ entnommen werden.

Die Schaltung mit dem ZLED7730 geht aus **Bild 1** hervor: Es handelt sich um einen *Continuous-mode Inductive Stepdown Converter*, er besteht aus IC1, D1, L1 und C3. Bei einem solchen Stepdown-Regler, auch *Buck Converter* genannt, ist die Ausgangsspannung stets niedriger als der Absolutwert der Eingangsspannung. Die Induktivität L1 und der Kondensator C3 bilden Energiepuffer, sie sorgen dafür, dass durch die Last, hier die LED, kontinuierlich ein konstanter Strom fließt.

Der Kondensator C1 glättet die Eingangsspannung, falls diese Spannung eine gleichgerichtete Wechselspannung ist. Da der Bordakku eine Gleichspannung liefert, ist der Wert von C1 nicht kritisch. Das Datenblatt der ZLED7030-Reihe [1] geht auf die Dimensionierung dieses Kondensators näher ein. C2 reduziert den so genannten *Ground Bounce Effect* [2], der immer dann auftritt, wenn der IC-interne, am Ausgang befindliche MOSFET schaltet. Dieser Kondensator muss möglichst nahe am IC montiert werden.

Der Wert der Induktivität L1 soll zwischen 30 µH und 220 µH liegen. Für Versuche mit anderen Ausgangsströmen kann abweichend von dem hier verwendeten Wert (100 µH) ein anderer Wert notwendig sein, damit die Schaltung stabil arbeitet. Höhere Werte passen infolge abweichender Bauformen eventuell nicht auf die Platine.

## Einstellen des Ausgangsstroms

Ein Merkmal der ZLED7030-Reihe ist die Eigenschaft, dass sich der Ausgangsstrom mit einem einzigen Widerstand $R_S$ einstellen lässt. Hier besteht dieser Widerstand wegen der feineren Abstufbarkeit aus den parallelen Widerständen R1 und R2. Für den Ausgangsstrom gilt folgende Beziehung:

$$I_{aus} = \frac{0{,}1V}{R_S}$$

Wenn R1 und R2 den Wert 1,2 Ω haben, beträgt der Wert der Parallelschaltung 0,6 Ω. Der Ausgangsstrom ist dann auf 166 mA eingestellt, für die Ikea-Leuchte „Jansjö“ ideal. Falls der Strombedarf des angeschlossenen Verbrauchers den Strom übersteigt, für den der Typ der ZLED-Stromquelle vorgesehen ist (siehe Tabelle 1), tritt die interne thermische Sicherung in Aktion.

**Tabelle 1. LED-Treiber ZLED7x30.**

| Typ | Ausgangsstrom |
|---|---|
| ZLED7030 | 1,2 A |
| ZLED7330 | 1,0 A |
| ZLED7530 | 750 mA |
| ZLED7730 | 350 mA |

## Einstellen der Dimm-Funktion

Das Verhalten des integrierten Dimmers hängt von zwei Lötbrücken oder ersatzweise zwei 0-Ω-Widerständen ab. In der Schaltung sind die Brückenkontakte mit *DIM 1* und *DIM 2* bezeichnet. Da beide Brückenkontakte einzeln entweder offen oder geschlossen sein können, sind vier

## STÜCKLISTE

**Widerstände:**
(SMD 0805, 1/4 W, 1 %)
R1,R2 = 1,2 Ω *

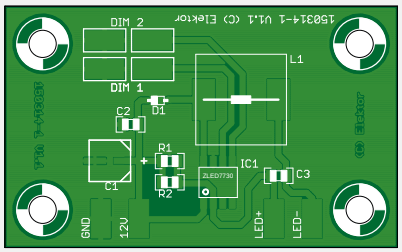
**Kondensatoren:**
C1 = 47 µF / 25 V SMD 0805 Aluminium-Elko (Panasonic EEE-FT1E470AR)
C2,C3 = 1 µF / 50 V X7R SMD 0805

**Induktivität:**
L1 = 100 µH SMD, 3,1 A / 0,11 Ω (Würth Elektronik 7447709101)

**Halbleiter:**
D1 = Schottky-Diode 30 V / 1 A SMD SOD-323F (NXP PMEG3010EJ)
IC1 = ZLED7730 LED-Driver SMD SOP-8 (IDT, ZMDI) *

**Außerdem:**
Platine 150314-1

* siehe Text



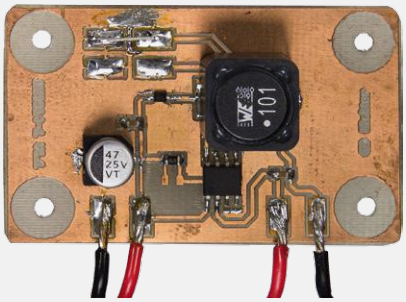



Bild 2. Die Platine für den LED-Treiber; sämtliche Bauelemente sind SMDs.

**Tabelle 2. Dimmer-Konfigurationen.**

| DIM1 | DIM2 | Dimm-Modus |
|---|---|---|
| Offen | Offen | Nicht dimmen: 100 % ➞ 0 % ➞ 100 % ➞ ... |
| Offen | Lötbrücke | 3 Stellungen: 100 % ➞ 50 % ➞ 20 % ➞ 100 % ➞ ... |
| Lötbrücke | Offen | 3 Stellungen: 100 % ➞ 60 % ➞ 30 % ➞ 100 % ➞ ... |
| Lötbrücke | Lötbrücke | 2 Stellungen: 100 % ➞ 30 % ➞ 100 % ➞ ... |

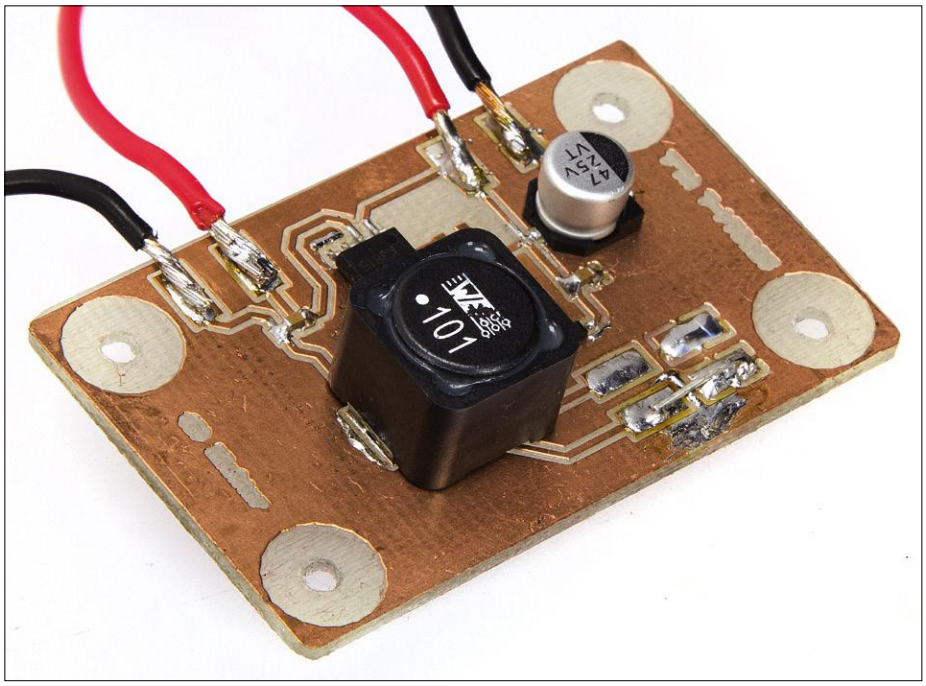


Bild 3. Die SMD-Bauelemente können noch von Hand montiert werden.

Konstellationen möglich. Die vier wählbaren Dimm-Varianten können **Tabelle 2** entnommen werden.
Das Umschalten der Dimm-Zustände, die mit dem Dimm-Modus festgelegt wurden, ist denkbar einfach: Es genügt, den LED-Treiber kurz aus- und wieder einzuschalten. Der Schalter ist in Bild 1 nicht dargestellt, er befindet sich in der Nähe der 12-V-Spannungsquelle.

### Auf- und Einbau

Für den 12-V-LED-Treiber haben wir eine kleine einseitige Platine entworfen, **Bild 2** zeigt, wie sie aussieht. Nebenbei bemerkt: Die Platine unseres Prototyps im Bild ist nicht geätzt, sondern gefräst. Sämtliche Bauelemente haben SMD-Bauform. Die Montage beginnt mit den passiven Komponenten, zuletzt folgt das IC.

> LEDs brauchen Stromquellen, nicht Spannungsquellen

Wenn Sie bisher noch keine Erfahrungen im Umgang mit SMDs sammeln konnten, bietet sich hier eine gute Gelegenheit. Das Fertigen von Prototypen ist nicht ohne gelegentliches Löten von SMDs mit der Hand möglich. Warum also nicht mit diesem Projekt einsteigen? Die hier verwendeten SMD-Komponenten sind noch nicht so winzig, dass sie nur mit einem SMD-Lötgerät wie zum Beispiel einem SMD-Lötofen montiert werden können. Ein Lötkolben mit einer bleistiftförmigen Spitze, eine Rolle dünner Lötdraht und ein kleiner Vorrat an Entlötlitze genügen. Eine Leselupe mit Beleuchtung ist eine wertvolle Hilfe, und eine ruhige Hand und etwas Geduld können auch nicht schaden! Überprüfen Sie nach der Montage die Platine mit einer Lupe auf eventuelle Zinnbrücken. Die aufgebaute Platine soll ungefähr so aussehen, wie in **Bild 3** gezeigt.
Der Autor hat ein Exemplar des 12-V-LED-Treibers in einem Mini-Gehäuse Typ 1551MBK von Hammond untergebracht. Im Gehäuse eingebaut ist die Schaltung mit einem Klettband auf der Fußklemme der Leuchte „Jansjö“ befestigt (**Bild 4**). Als Schalter kann der originale Schnurschalter der Leuchte dienen, oder im

## Glühlampen und moderne LEDs

Die klassischen Glühlampen, die nach dem Willen der EU bald nur noch im Museum anzutreffen sind, können ihre lichtspendende Wirkung an der Netzspannung 230 V entfalten, weil der Widerstand des Glühwendels den hindurchfließenden Strom begrenzt. Der Widerstand steigt mit zunehmender Temperatur, der Temperaturkoeffizient ist folglich positiv. Kurz nach dem Einschalten ist ein Gleichgewicht erreicht, bei der die Glühlampe ihre nominale Helligkeit erreicht hat. Die modernen LEDs verhalten sich genau betrachtet umgekehrt. Eine LED ist eine Halbleiterdiode, an der im Betrieb eine bestimmte Spannung, die Durchlassspannung, abfällt. Der Strom wird nicht von der LED begrenzt, im einfachsten Fall wird dazu ein Widerstand vorgeschaltet. An diesem Widerstand, der für unterschiedliche Konfigurationen stets anders dimensioniert werden muss, fällt eine Spannung ab. Der Spannungsabfall ist genau so hoch, dass durch die LED ein Strom fließt, bei dem die LED mit nominaler Stärke leuchtet. Wenn die Ikea-Leuchte unmittelbar an die Spannung 12 V gelegt würde, findet keine Strombegrenzung statt. Die LED würde für einen kurzen Moment grell aufleuchten, um sich danach in die ewige Dunkelheit zu verabschieden – sie wäre defekt. Der Strombegrenzungswiderstand setzt abhängig von der anliegenden Spannung mehr oder weniger viel elektrische Energie in nutzlose Wärme um, was nicht besonders effizient ist. Die sicher klügere Idee besteht darin, einen elektronischen LED-Treiber vorzuschalten, der sich wie eine Stromquelle verhält. Er sorgt dafür, dass durch die Leuchte ein Strom fließt, der im sicheren Bereich der LED liegt.

Gehäuse wird ein von außen bedienbarer Schiebe- oder Kippschalter angebracht. Die Platine dieses Prototyps weicht im Bild erkennbar von der Platine ab, die im Elektor-Labor entworfen wurde und im Elektor-Shop erhältlich ist. ⏮

(150314)gd

Anmerkung der Redaktion:
Ikea bietet auch eine Version „Jansjö/USB“ (ohne 230-V-Adapter, ohne Kabel und Schnurschalter, mit USB-Stecker) für rund 5 € an.

### Weblinks

[1] www.idt.com/document/dst/zled7x30-datasheet

[2] https://de.wikipedia.org/wiki/Ground_bounce

**EINKAUFSZETTEL**

→ 150314-1
Platine

Bild 4. In ein Gehäuse eingebaut kann die Schaltung auf der Fußklemme der Leuchte befestigt werden (im Bild ist der Prototyp des Autors zu sehen).

Anzeige

HAMMOND MANUFACTURING®

Hand Held Gehäuse
Standard und Wasserdicht

www.hammondmfg.com/1553.htm
www.hammondmfg.com/1553W.htm

01256 812812
sales@hammondmfg.eu

# (Fast) alles, was Sie schon immer wissen wollten über ...

# die Qualität von Audio-Steckern

Antworten von **Rémy Mallard** (F)

Vom alten, verrosteten Original-DIN-Stecker bis zur gefälschten, goldglänzenden XLR-Version: Audiostecker offenbaren sich in verschiedenen Formen und Qualitäten. Wir haben Rémy Mallard gebeten, uns mehr darüber zu erzählen.

*Foto : Shutterstock*

**Q** *Warum gibt es so viele Stecker-Formate?*

**A** Der Bedarf an Steckern wächst im professionellen und privaten Bereich immer weiter. Die Industrie entwickelt immer „leistungsfähigere und zukunftsweisendere" Produkte mit stets neuen Funktionen, die ständig neuen elektrischen Anforderungen wie zum Beispiel CEM unterliegen (abhängig von der Art und Geschwindigkeit der Signale) oder sich nach neuen mechanischen Anforderungen richten müssen (immer mehr Kontakte und Konfektionierungen für stationäre und mobile Anwendungen). Nach dem dreipoligen DIN-Stecker und dem einpoligen RCA/Cinch-Typ ist heute die Zeit der 29-poligen HDMI-Verbindungen, die **auch** Video und Internet unterstützen.

**Q** *Analog und Digital: Kein Unterschied?*

**A** Egal ob analog oder digital: Die Qualität der Kabel und Stecker ist wichtig. Im analogen Bereich kündigt sich ein Defekt bereits vorher durch Verschlechterung der Qualität und Knistergeräusche an. Bei digitalen Systemen offenbaren sich die Störungen in wesentlich gemeinerer Weise, denn der Übergang zwischen „alles gut" und „nichts geht mehr" vollzieht sich hier ganz plötzlich. Aus der Menge der durch die Stecker fließenden Daten und der damit verbundenen Taktfrequenz ergeben sich Probleme hinsichtlich der Impedanz und der Länge sowie der parasitären Kapazitäten der Kabel und der Kontakte. Mit normalen Kabeln und irgendwelchen Steckern ist es nicht mehr getan: Hier geht es um Kabel-Impedanzen (z.B. 50, 75 oder 110 Ω) und die dazu passenden Stecker. Die digitale, symmetrische AES-Verbindung [1] benötigt zum Beispiel ein Kabel mit einer Impedanz von 110 Ω mit XLR-Steckern, und eine digitale, asymmetrische S/PDIF-Verbindung erfordert ein Kabel von 75 Ω mit 75-Ω-BNC-Steckern (im privaten Bereich kommen oft auch die preiswerteren RCA/Cinch-Varianten zum Einsatz). Bei HDMI fließen die Daten im TMDS-Format [2] über verdrillte Leitungen, deren differentieller Widerstand 100 Ω beträgt. Bei einem theoretischen Durchfluss von 48 Gigabit pro Sekunde bei HDMI 2.1 ist die Qualität von Steckverbindung und Kabel sogar noch wichtiger.

**Q** *Worin bestehen die Unterschiede zwischen professionellen Steckern und Steckern für die private Anwendung?*

**A** Ein Stecker für die Massenanwendung muss preisgünstig – und ein professioneller Stecker muss zuverlässig sein: In Tonstudios und Sendeanstalten erwartet man von professionellen, festen Kabelverbindungen eine gewisse Langlebigkeit, und der Alterungsprozess darf nicht hörbar sein. Kabel und Stecker schlechter Qualität altern viel schneller (Oxidation und/oder dielektrische Alterung). Stecker, die z.B. bei Konzerten zum mobilen Einsatz kommen, dürfen nicht zerbrechen, wenn man versehentlich auf sie tritt - und müssen in ihren Buchsen stecken und unbeschädigt bleiben, wenn einmal jemand über ein Kabel stolpert. Kurz und gut: Die Stecker müssen robust und daher besser aus Metall statt aus Plastik gefertigt sein, was auch für den Amateurbereich empfehlenswert ist – auch bei DIN- oder RCA/Cinch-Steckern. Ein mit Masse verbundenes, leitendes Steckergehäuse schützt nicht zuletzt auch vor Störeinstrahlungen. Mit verschiedenen Methoden wie Laschen, verschraubbaren Schalen oder gezahnten Klammern kann das Kabel vor dem Abreißen geschützt werden. Doch Vorsicht bei den beiden letztgenannten Möglichkeiten, deren Wirkung nur bei bestimmten Kabeldurchmessern garantiert werden kann.

**Q** *Stecker-Plagiate: Gibt es sie wirklich?*

**A** Welche bekannte Marke bleibt nicht von Plagiaten verschont! Wer freut sich nicht über einen XLR-Stecker der

Foto 1. Der gefälschte Neutrik-Stecker passt zwar gut in die entsprechende Original-Buchse, aber die Verriegelung (eines der Hauptmerkmale dieses Steckertyps) funktioniert nicht.

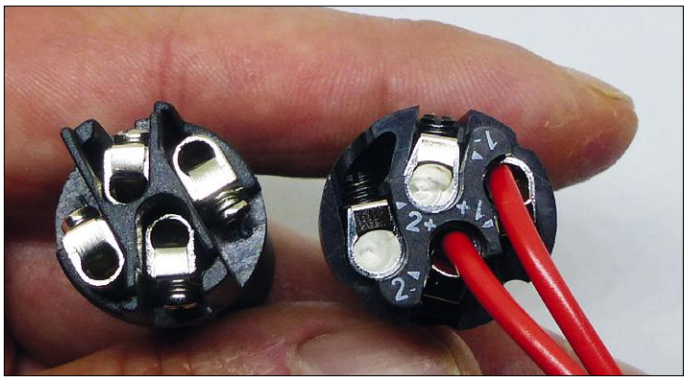



Foto 2. Links eine Neutrik-Fälschung, rechts ein Original-Amphenol-Stecker. Bei der Kopie scheint die Lasche zum Festhalten des abisolierten Kabels jedoch eher das Gegenteil zu bewirken.

Marke Neutrik, der 2 bis 5 Mal preiswerter als beim Händler um die Ecke ist. Auch wenn unser Bauchgefühl Zweifel anmeldet (bei dem Preis kann doch etwas nicht stimmen), lassen wir uns seltsamerweise oft von falschen Etiketten bekannter Marken in die Irre führen. Und je billiger etwas ist, desto geringer ist schließlich auch das finanzielle Risiko, falls man doch hereinfällt. Während der Amateur sich solch ein Verhalten leisten kann, muss der Profi sehr lange nachdenken, bevor er nach dem Motto „Billig um jeden Preis“ handelt. Im Laufe der Zeit haben die renommierten Hersteller ihre Stecker mit zahlreichen mechanischen und elektrischen Verbesserungen ausgestattet, wogegen dies bei billigen Kopien nicht der Fall ist. Auf den ersten Blick ist bei einer Fälschung oft alles in Ordnung. Erst später erfolgt die Ernüchterung, zum Beispiel, wenn man den Stecker demontieren und für andere Zwecke verwenden möchte und dann erkennt, dass sich Qualität am Ende doch bezahlt macht. Die gefälschten Stecker mögen zwar genauso „sexy“ aussehen wie die Originale, aber sie sind nicht so zuverlässig!

**Q** *Bergen gefälschte Stecker Gefahren?*

**A** Solange es sich nur um schwache Signale (z.B. Line oder Mikrofon) handelt, beschränkt sich das Risiko auf sporadische Knackgeräusche, Pegelabsenkungen oder Tonausfälle. Vor allem störend ist das, wenn mit diesen Signalen eine Verstärker-Anlage von mehreren kW gespeist wird. Wenn in den Steckern jedoch höhere Ströme fließen (z.B. Speakon-Stecker für Beschallungssysteme) ist das Risiko wesentlich höher (starke Erhitzung). Marken-Stecker für hohe Leistungen garantieren nicht nur den kleinstmöglichen Übergangswiderstand, sondern auch eine gleichmäßige Kontaktoberfläche zur Vermeidung punktueller Erhitzung, die zur Verkohlung von Kunststoffteilen führen kann. Diese Garantie muss auch in hektischen und stressigen Live-Situationen gegeben sein, wo Stecker sehr häufig und nicht gerade mit Feingefühl eingesteckt und herausgezogen werden.

**Q** *Lohnt es sich noch, Kabel selber anzufertigen?*

**A** Ich persönlich befinde mich irgendwo in der Mitte zwischen den supergünstigen Wegwerf-Kabeln unter einem Euro und den High-Quality-Versionen für 1000 € oder mehr, die sich nicht wiederverkaufen lassen. Bei einfachen Steckern ist die Anfertigung eigener Kabel durchaus möglich und lohnenswert, bei komplizierten Steckern wie HDMI so gut wie unmöglich, was jeder versteht, der schon einmal einen geöffnet hat. Die Anfertigung eigener Kabel hat den Vorteil, dass man Kabel und Stecker unabhängig kombinieren und später auch wieder leichter reparieren kann. Die durch ungeeignete Kabel bedingten Alterungsprozesse können in bestimmten Fällen natürlich schon die Klangqualität verschlechtern - bei analogen Kabeln (Höhenabschwächung) ist das tolerierbar, bei digitalen Kabeln inakzeptabel (Klickgeräusche oder Ausfälle). Man sagt, dass ein S/PDIF-Kabel von einem Meter Länge, das aus Standard-Abschirmkabel hergestellt ist und zwei RCA-Stecker „non 75 Ω“ besitzt, die Mitten unterdrücken soll. Doch Probieren geht über Studieren. ⏮

(160471)

## Weblinks

[1] AES3, Kommunikationsstandard für digitale Audiodaten: https://en.wikipedia.org/wiki/AES3

[2] TMDS, Übertragungsnorm für Videodaten: https://en.wikipedia.org/wiki/transition-minimized_differential_signaling

**Über den Autor**

Rémy ist ein passionierter Audio-Liebhaber, der mit 14 Jahren seinen ersten Radiosender gebaut hat und seinen ersten Mini-Synthesizer mit 15 Jahren zusammenlötete (der *Formant* war zu dieser Zeit noch zu groß für ihn). Er hat 23 Jahre lang beim Fernsehsender TDF (Télédiffusion de France) in der Abteilung *Satellitenfernsehen* gearbeitet, um sich dann mit Fernseh- und Radio-Sendeanlagen sowie mit Rundfunkstudios zu beschäftigen (Wartung, Gutachten, Schulungen). Heute konzipiert er elektronische Schaltungen für den Audiobereich und führt professionelle Audio-Schulungen durch. Er arbeitet außerdem als Off-Sprecher und analysiert und bewertet Tonbänder. Auf seiner Webseite *sonelec-musique.com* stellt er seit 2004 seine Arbeit in humorvoller Weise vor.

# GoNotify, ein flexibles IoT-Sensorinterface

## Einfach in die Cloud

Von **Gino De Cock** (Belgien)

**Wir hören viel über IoT = das Internet der Dinge, IIoT, vernetzte Sensoren und Kühlschränke.... In ein paar Jahren soll es einen riesigen Markt und Milliarden von vernetzten Geräten geben, aber Sie gehen immer noch zu einem Wandschalter, um das Licht im Wohnzimmer anzuschalten? Jetzt sind die dunklen Tage aber vorbei, denn mit GoNotify kann man ans Internet anschließen, was einem gefällt.**

Haben Sie sich jemals gefragt, was in und um Ihr Zuhause geschieht, wenn Sie nicht dort sind? Sind die Kinder drinnen oder draußen? Haben Sie das Haus in Brand gesetzt? Sind die Heizung an und der Wasserhahn wirklich zugedreht? Oder setzt er gerade die Küche unter Wasser? Ist vielleicht jemand an der Tür? Möchten Sie Ihrem Hund auf Facebook folgen? Oder umgekehrt: Sie sind zu Hause, aber in der Fabrik geht irgendetwas schief. Eine Maschine überhitzt oder das Kühlgerät kühlt nicht kühl genug? Fließt das Wasser dort, wo es fließen sollte? All dies kann kontrolliert werden, indem man den richtigen Sensor an der richtigen Stelle anbringt und ihn mit GoNotify mit dem Internet verbindet. So einfach ist das!

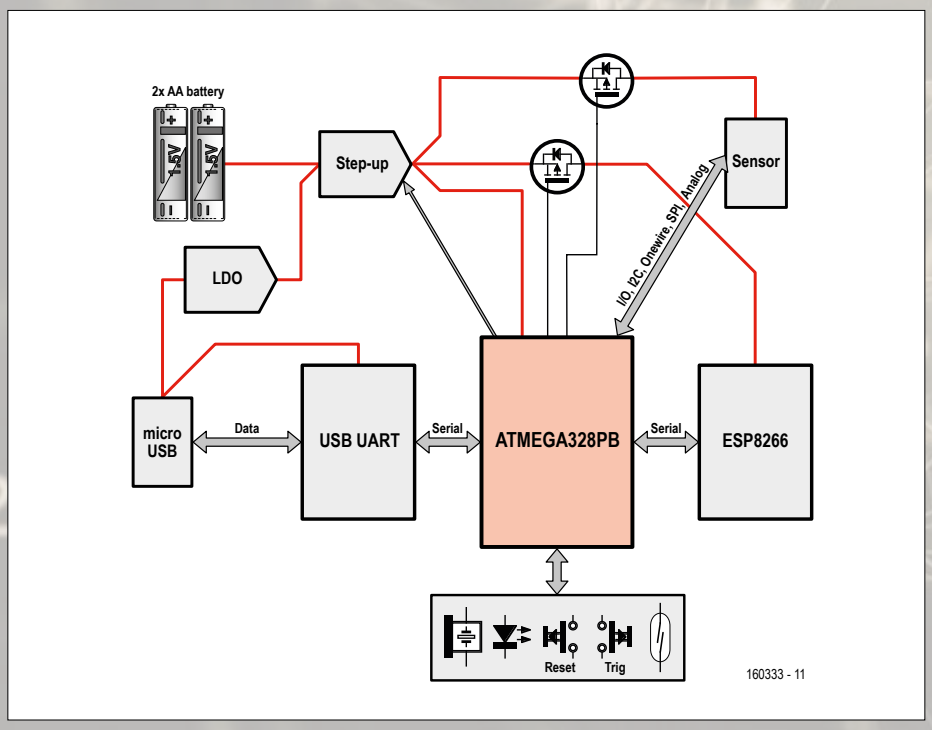



Bild 1. Die Funktionsübersicht von GoNotify.

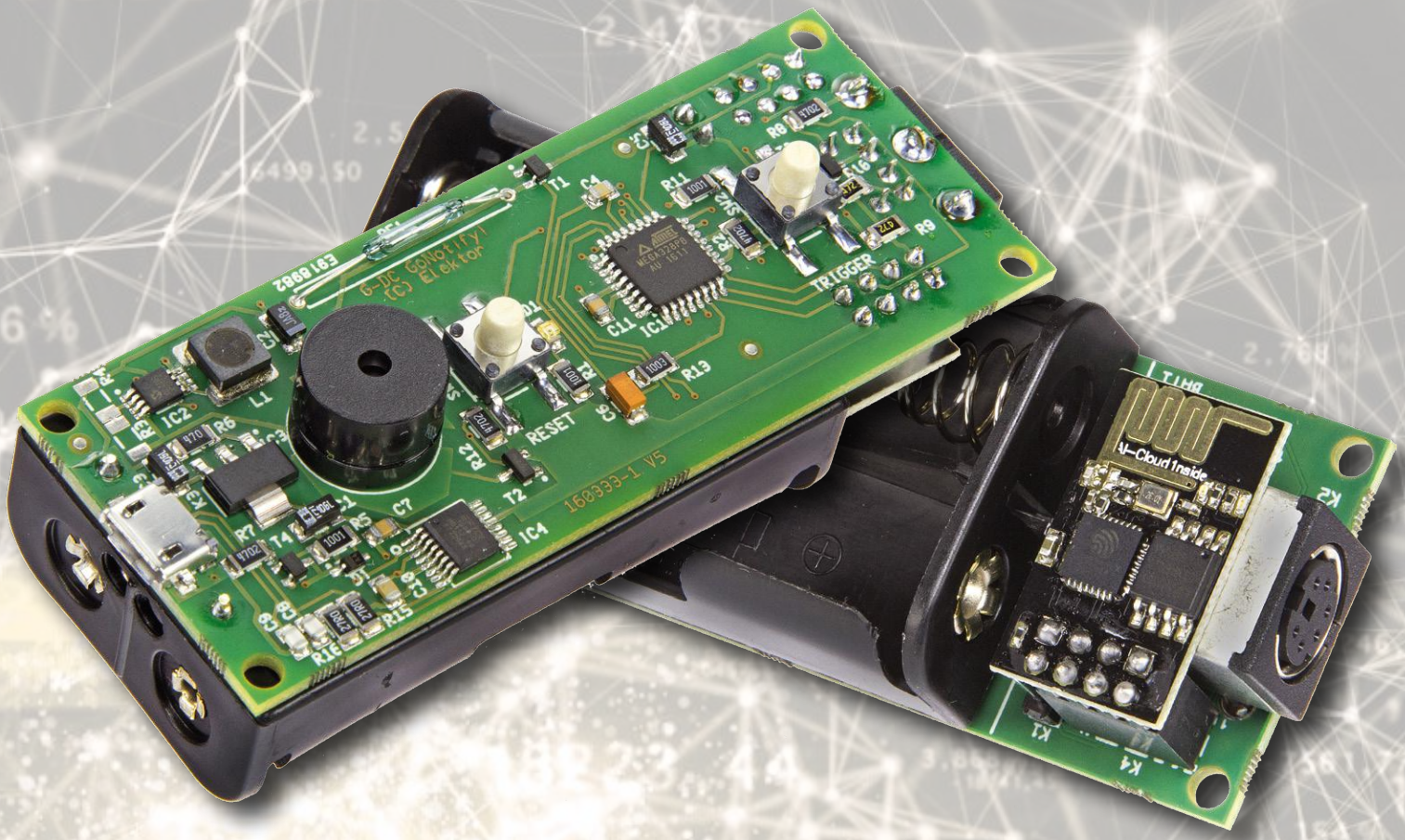
### Der Anfang

Vor einiger Zeit fragte ich mich, was die Voraussetzungen für ein vernetztes Gerät wären. Es musste drahtlos sein, also mit Batterien oder Solarenergie betrieben werden. Dies bedeutet, dass es mit so wenig Energie wie möglich auskommen, also ein Low-Power-Design sein muss. Darüber hinaus sollte das System einfach und kostengünstig sein, auch was die Installation betrifft. Und schließlich muss das Gerät flexibel und einfach zu entwickeln sein.

Mit dem beliebten und preisgünstigen WLAN-Chip ESP8266 kann die „Drahtlosigkeit" erreicht werden, allerdings könnte dabei die Low-Power-Bedingung eine Herausforderung darstellen. Mit all dem im Hinterkopf habe ich mich an die Arbeit gemacht.

### Die Hardware

Der resultierende Entwurf besteht aus vier Hauptblöcken (**Bild 1**):




- für den Internet-Anschluss ein billiges ESP-01S-Modul mit dem ESP8266-WLAN-Mikrocontroller
- ein ATmega328PB für den Anschluss der Sensoren und die Umrechnung der Daten
- ein USB-Port für Programmierung und Debugging
- ein Leistungsmodul, bestehend aus einem Low-dropout- und einem Step-up-Regler.

Die Schaltung ist in **Bild 2** zu sehen. Der Mikrocontroller ist ein ATmega328PB, eine aktualisierte Version des beliebten ATmega328P mit vielen neuen Eigenschaften. Eine davon ist die reduzierte Leistungsaufnahme, aber es gibt auch einen genaueren internen RC-Oszillator, der einen externen Quarz überflüssig macht. Am wichtigsten ist, dass der Controller zwei serielle Schnittstellen besitzt. Serial0 wird hier zum Hochladen der Firmware und zum Debuggen verwendet, über Serial1 verläuft die Kommunikation mit dem ESP-01S-Modul. Die Hauptaufgabe des ATmega328PB ist es, die Sensoren auszulesen, und dies mit so wenig Energie wie möglich.
In dieser Schaltung wird der WLAN-Chip ESP8266 auf einem Modulplatinchen verwendet, das im Internet unter dem Namen ESP-01S leicht und sehr preiswert erhältlich ist. Das Modul wird an K4 angeschlossen. Damit ist es mit der seriellen Schnittstelle 1 (Serial1) der MCU und zwei seiner GPIO-Pins verbunden. Die Stromversorgung erfolgt über Transistor T1, der von der MCU (also dem ATmega) ein- und ausgeschaltet wird. Es ist wichtig, die 3,3-V-Stromschiene vor dem Einschalten von T1 zu aktivieren, um eine zuverlässige Verbindung zum Internet zu gewährleisten.
Wenn der ESP-01S eine neue Firmware für den ATmega328PB over-the-air (OTA) empfängt und sie zur MCU leiten möchte, muss zuvor ein Reset der MCU erfolgen, damit sie auch im Bootloader-Modus startet. Um die Stromversorgung des ESP-01S-Moduls auch während des Reset-Vorgangs zu gewährleistet, wird T1 statt von der MCU von R13 und C6 für eine kurze Zeit aktiviert.
IC4 ist eine USB-zu-UART-Brücke namens FT230XS. Sie bietet einen komfortablen Programmier- und Debugging-Port an der seriellen Schnittstelle Serial0 der MCU. Die fallende Flanke des RTS-Signals wird von C4 und R11 extrahiert und als Reset-Signal für die MCU verwendet, so dass sie Firmware von der Arduino-IDE empfangen kann.
IC3 verwandelt die 5 V des USB in 3,3 V.

## INFOS ZUM PROJEKT

ATmega328PB | ESP8266 | IoT | AllThingsTalk | IFTTT | FFA 2016

Einsteiger
➡ **Fortgeschrittene**
Experte

etwa 4 Stunden

SMD-Löten,
PC,
Arduino-IDE

etwa 30 €

## Eigenschaften

- Universelles drahtloses Sensorinterface
- Unterstützt WLAN und ESP-Now
- Ultra low Power
- Arduino-kompatibel
- Grove-kompatibel

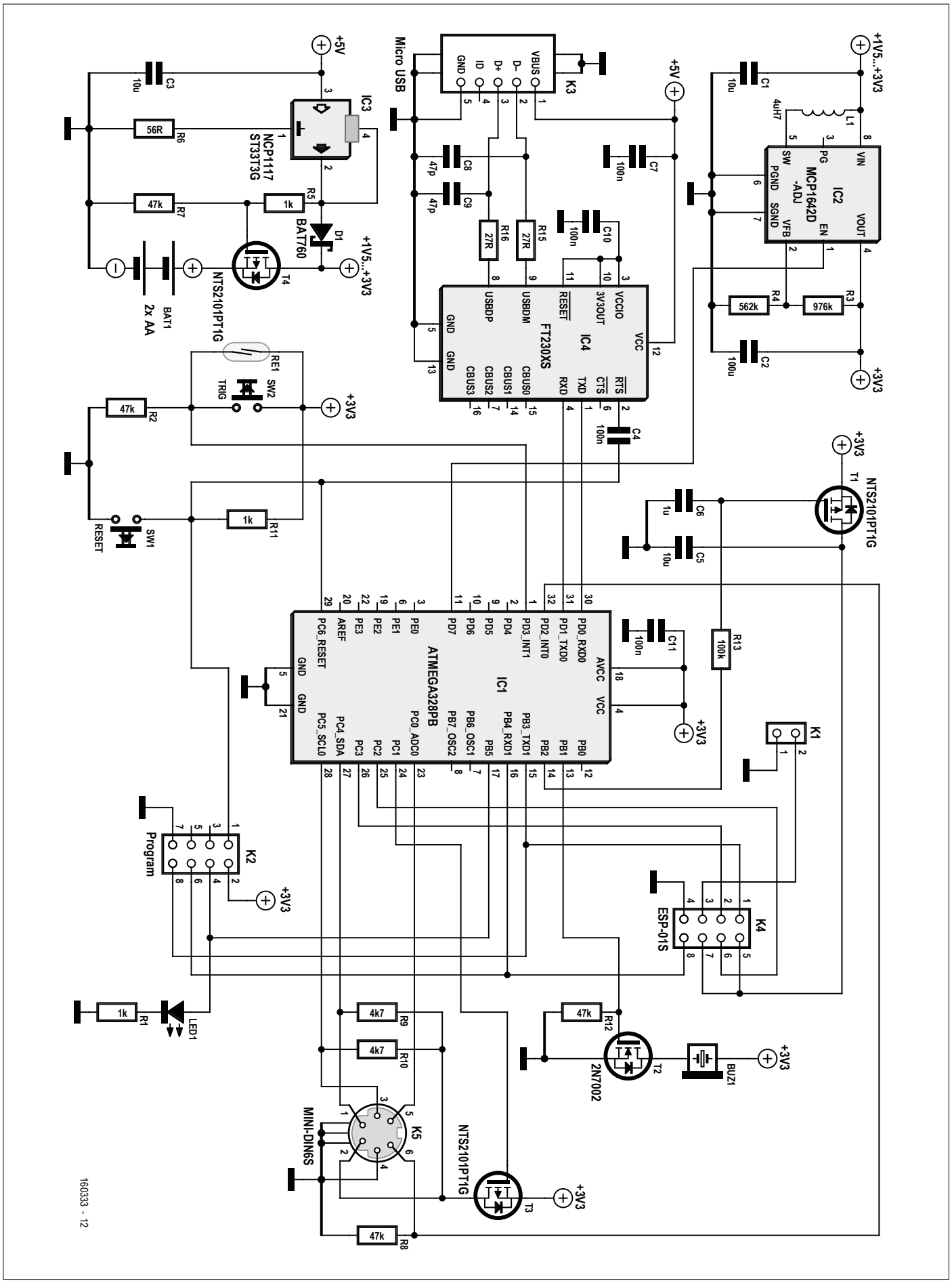



Bild 2. Vollständige Schaltung des Geräts. Das WLAN-Modul ist an K4 angeschlossen.

Tatsächlich ist die Ausgangsspannung von IC3 wegen R6 etwas höher als 3,3 V, um den Verlust an der Schottky-Diode D1 zu kompensieren.
Die Schaltung wird unabhängig von ihrem Status (das heißt, aktiviert oder nicht) über IC2 versorgt. Wenn der Enable-Eingang auf Low gehalten wird, befindet sich der Regler im Bypass-Modus, so dass die Ausgangs- stets der Eingangsspannung folgt. IC2 ist in mehreren Varianten erhältlich. Hier wird der Adjustable-Typ für größtmögliche Flexibilität eingesetzt, es ist aber auch der Festspannungsregler MCP1642D-33I/MS 3.3 V möglich. In diesem Fall lassen Sie R3 und R4 weg. Verwenden Sie aber keinesfalls eine B-Version (MCP1642B- ...) ohne Eingangs-Bypass-Modus.
Die Sensoren werden an der sechspoligen Mini-DIN-Buchse K5 angeschlossen. Die Platine ist so ausgelegt, dass Sie auch eine 4-polige Grove-Buchse von Seeed Studio montieren können.
Wenn es an der Zeit ist, einen Sensor auszulesen, aktiviert der Controller über T3 zunächst die Stromversorgung des Sensors. Wenn der Sensor 3,3 V benötigt, wird auch der Step-up-Converter IC2 aktiviert (dies muss natürlich vom Programmierer der Applikation, das heißt von Ihnen, so programmiert werden). Mit Reed-Schalter RE1 und Taster SW2 (Trigger) lässt sich die Schaltung testen, auch wenn kein externer Sensor angeschlossen ist. Sie wecken die MCU über den Interrupteingang INT1.
K2 ist für die In-Circuit-Programmierung der MCU (zum Beispiel mit einem Custom-Bootloader) vorgesehen. Ein Tastendruck auf SW1 setzt die MCU zurück. K1 ermöglicht den Zugriff auf den zweiten seriellen Port des ESP8266-Mikrocontrollers. Dies kann nützlich sein, um Ihren WLAN-Code zu debuggen, ohne die Kommunikation mit der MCU zu unterbrechen. Die Allzweck-LED1 nimmt die Rolle der berühmten „Arduino-Pin13-LED“ ein. Ihr kann eine beliebige Funktion zugeordnet werden.

## Power-Management

Wenn man die Schaltung mit zwei AA-Batterien versorgt, ist es möglich, im „guarding mode“ die Stromaufnahme auf nur 10 µA zu drücken. In diesem Low-Power-Modus wird so wenig Schaltung wie möglich aktiviert, ohne die Sensorüberwachung zu stoppen. Der ESP-01S ist ausgeschaltet, der Step-up-Wandler befindet sich im Bypass-Modus und der Controller schläft. Nur der Watchdog-Timer des Controllers ist aktiv und weckt periodisch die MCU, damit ein Sensor ausgelesen wird. Dies kann auch mit einem externen Interrupt erreicht werden.
Im Guarding-Modus sind das ESP-01S-Modul über T1 ausgeschaltet und der Step-up-Regler IC2 deaktiviert, indem der Enable-Pin auf Low gelegt wird. IC2 befindet sich dann eigentlich im Bypass-Modus, in dem der Ausgang mit dem Eingang verbunden ist. Die internen Funktionen sind aber deaktiviert, so dass die Stromaufnahme verschwindend gering ist. Nur der ATmega328PB (IC1) wird mit Strom versorgt, da er entweder über T4 mit den Batterien oder, wenn 5 V an Pin 1 der Micro-USB-Buchse K3 anliegt, über D1 mit dem Low-Dropout-Regler IC3 verbunden ist.
Mit reduzierter HF-Sendeleistung und so kurz wie möglich gehaltenen Internet-Verbindungen verbraucht GoNotify im Durchschnitt etwa 1,5 mAh pro Nachricht, auch wenn ein REST-Protokoll über eine sichere HTTPS-Verbindung verwendet wird.
Bei der Stromversorgung mit zwei frischen 2500-mAh-Batterien und 1,5 mAh pro Nachricht kommt man auf 3.000 bis 4.000 Nachrichten. Geht man von einer Nachricht pro Stunde aus, reichen die Batterien etwa fünf Monate. Es ist also lohnend, einige Intelligenz im Mikrocontroller unterzubringen, um die Zahl der Nachrichten auf ein Mindestmaß zu reduzieren. Wenn ein Fehler erkannt wird, kann GoNotify den (lokal anwesenden) Nutzer auch mit Summer BUZ1 (eingeschaltet von T2) ohne Internetverbindung informieren.
Wenn eine Anwendung, zum Beispiel ein MQTT-Client, eine ständige Internetverbindung benötigt, sollte man GoNotify über den Micro-USB-Anschluss (K3) versorgen. In diesem Fall stellen die Batterien eine Back-up-Stromversorgung dar, wenn die USB-Stromversorgung einmal ausfällt.
Um die Anzahl der Nachrichten pro Batterieladung zu maximieren, stehen folgende Optionen zur Verfügung (abgesehen vom Schreiben cleverer Software):

- Reduzieren Sie die Sendeleistung! Der ESP-01S arbeitet standardmäßig mit einer ziemlich hohen Reichweite und benötigt beim Senden eine Menge Power. Je nach Situation kann die Reichweite reduziert und damit Batteriekapazität eingespart werden. Dies kann durch Software erfolgen, aber auch mit dem „Flash Download Tool“ [1] für den ESP8266 auf der Registerkarte RFConfig.
- Legen Sie den WLAN-Kanal fest! Das Scannen von WLAN-Kanälen kostet Energie. Dazu muss man dem ESP-01S-Modul mitteilen, dass es denselben WLAN-Kanal wie der Access Point verwenden soll.
- Vermeiden Sie DHCP! DHCP-Verhandlungen können durch die Festlegung der IP-Adresse vermieden werden (auf Kosten der Flexibilität natürlich).
- Vermeiden Sie DNS-Lookup! Verwenden Sie die IP-Adresse des Ziels (Cloud oder andere) statt des URL, dies spart Zeit und Strom.
- Optimieren Sie den HF-Initialisierungsprozess des ESP-01S! Normalerweise führt der ESP-01S beim Start eine HF-Ausrichtung durch, die auch ein wenig Strom verbraucht. Dies geschieht mit der API-Funktion `system_phy_set_powerup_option(2)`. Auch das „Flash Download Tool“ kann dieses Verhalten ändern, allerdings ist dort die beste Option (Nr. 2) nicht verfügbar.
- Reduzieren Sie die Taktfrequenz der MCU, indem Sie die Fuse DIV2 setzen. Dann läuft die MCU mit 4 MHz anstelle von 8 MHz, so dass die MCU läuft, bis die Batteriespannung unter 1,8 V fällt.

## Gehäuse

Neben der Schaltung, einer Platine und viel Software habe ich auch ein 3D-druckbares Gehäuse für GoNotify erstellt (**Bild 3**). Sie können es zu Hause oder online drucken, wenn Sie einen Service in Ihrer Nähe finden. Die Design-Dateien für das Gehäuse können wie alle anderen GoNotify-Dateien von [2] heruntergeladen werden.

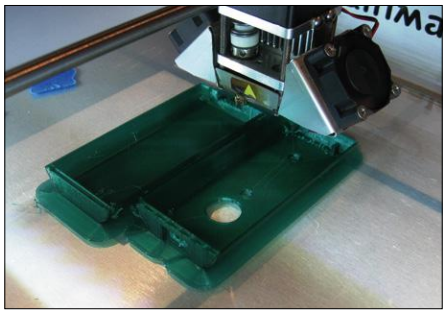
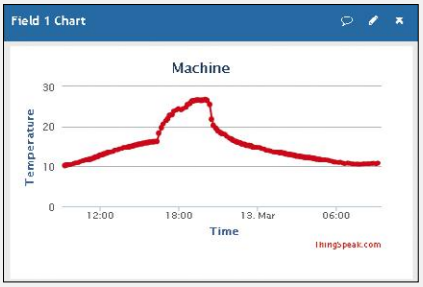
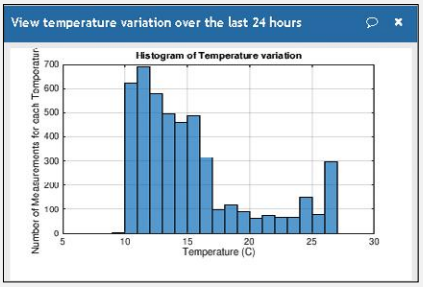


Bild 3. Ein Design für ein 3D-druckbares Gehäuse gibt es auch.

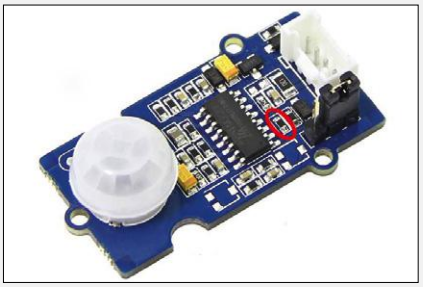
### Statistische Online-Analyse mit ThingSpeak




Überwachen Sie die Temperatur einer Maschine mit einem geeigneten Sensor (zum Beispiel einem Thermoelement). Verbinden Sie GoNotify mit dem Online-Service ThingSpeak und analysieren Sie die erfassten Daten mit Matlab.
Source: https://goo.gl/tthgeJ
ThingSpeak: https://thingspeak.com

### Bewegungserkennung mit IFTTT

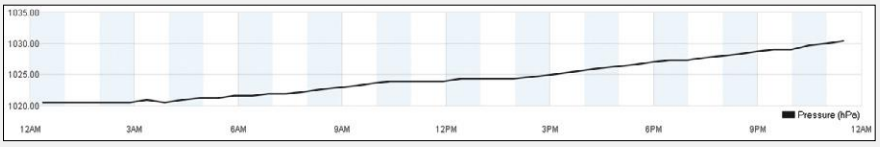


Den eingekreisten Widerstand kurzschließen.

Überwachen Sie Bewegung und werden Sie über den beliebten Online-Service „If This Then That“ (IFTTT) alarmiert. Um einen Grove-PIR-Bewegungssensor aus dem SeeedStudio mit GoNotify zu verbinden, muss man den 10-kΩ-Widerstand in der Ausgangsleitung des Sensors (siehe Foto) kurzschließen, da er sonst die 4,7-kΩ-I2C-Pull-up-Widerstände des GoNotify stören würde. Verbinden Sie Ihr IFTTT-Applet mit einer Google-Kalkulationstabelle und protokollieren Sie die Bewegungen.
Source: https://goo.gl/shSWe6
IFTTT: https://maker.ifttt.com

### Werden Sie Teil des Weather-Underground-Netzwerks

Mit einem BMP180-Sensor am GoNotify werden Sie Teil des Weather-Underground-Netzwerks und stellen qualitativ hochwertige Wetterinformationen für jedermann auf diesem Planeten zur Verfügung.
Source: https://goo.gl/A3QByX
Weather Underground: www.wunderground.com




### Update der Firmware

Dank eines kompatiblen Bootloaders kann die Firmware der MCU über USB mit der Arduino-IDE aufgespielt werden. Außerdem ist über K2 eine In-Circuit-Programmierung mit einem AVR-Programmer möglich. Um jedoch das ESP-01S-Modul über die USB-Verbindung zu programmieren, ist ein ausgeklügelter Mechanismus erforderlich: Die MCU muss dazu den ESP-01S zuerst in den Bootloader-Modus überführen und dann eine Brücke zwischen den beiden seriellen Ports herstellen, um die neue Firmware an den ESP-01S weiterzureichen. Dies wurde durch eine individuelle Anpassung des MCU-Bootloaders erreicht.

Die Datenrate beim Programmieren des ESP-01S ist auf 57.600 Bits/s begrenzt. Eine schnellere Programmierung ist aber trotzdem möglich, wenn Sie einen Over-the-Air-Code in die Sensoranwendung aufnehmen. Das ESP-01S-Modul verfügt über 8 MBit Flash-Speicher, das Minimum, das für OTA-Updates benötigt wird. Das Netzwerk R13/C6 ist für OTA-Firmwareupdates des ATmega328PB gedacht. Wenn der ESP-01S die neue Firmware empfängt und sie zum ATmega328PB schiebt, muss ein Reset des Controllers ihn in den Bootloader-Modus überführen. Während dieses Vorgangs muss aber der ESP-01S weiterlaufen. Dafür zeichnen R13 und C6 verantwortlich.

### Applikationsentwicklung

Für praktische Anwendungen ist es am besten, die Aufgaben zwischen den beiden Mikrocontrollern aufzuteilen. Der ATmega328PB überwacht den Sensor, löst bei Bedarf einen Alarm aus und aktiviert das ESP-01S-Modul. Dieses sorgt dann für eine sichere Verbindung zum Internet und handelt die Kommunikation nach einem Protokoll Ihrer Wahl (HTML, REST, MQTT ...) ab. Der ESP-01S kann auch die Benutzeroberfläche (UI) für die Konfiguration oder die Überwachung über einen Standard-Internet-Browser auf dem Computer oder Smartphone zur Verfügung stellen.
Damit die Entwicklung der Applikation schön einfach bleibt, sollten auch die Programmierung und das Debuggen der Sensoranwendung des Gerätes einfach sein. Ermöglicht wird dies durch die Arduino-IDE.

In der Arduino-IDE müssen im Menü „Datei“ unter „Voreinstellungen“ im Feld

## Über den Autor

Gino De Cock (1977) ist von seiner Jugend an von Elektronik fasziniert. Er studierte Elektronik in Gent (Belgien) und graduierte dort 1999. Gepeinigt von der existenziellen Frage: „Gibt es keinen besseren Weg?“ verwandelt Gino seine Ideen in Projekte wie GoNotify.

**Fast Forward Award 2016**

GoNotify war einer der Beiträge zum Fast Forward Award, der von Elektor zusammen mit der Messeleitung der electronica im Jahr 2016 in München ausgelobt wurde. Der FFA war eine großartige Gelegenheit für Maker, Start-ups und Profis, ihre Projekte zu präsentieren.

„Zusätzliche Boardverwalter-URLs“ die URLs zweier Board-Definitionsdateien hinzugefügt werden. Man kann mehrere, durch Kommas getrennte URLs in einem Rutsch hinzufügen (es gibt keine Leerzeichen in den folgenden URLs):

- Für den ATmega328PB: https://raw.githubusercontent.com/ginodecock/V3GoNotify/master/ArduinoBoard/package_gonotify_v3_index.json
- Für den ESP-01S: http://arduino.esp8266.com/versions/2.3.0/package_esp8266com_index.json

Das Programmieren mit der Arduino-IDE ist ganz einfach:

- Verbinden Sie GoNotify mit dem PC. Beim ersten Mal wird ein virtueller COM-Port eingerichtet.
- Installieren Sie die ATmega328PB- und ESP-01S-Board-Definitionsdateien in der Arduino-IDE:
  Öffnen Sie im Menü „Werkzeuge/Board“ den Boardverwalter, wählen Sie „Beigetragen“ und installieren Sie „GoNotify-V3 ATMEGA“.
  Dasselbe führt man für den ESP-01S durch. Dazu wird „esp8266 by ESP8266 Community“ installiert.
- Wählen Sie im Menü „Werkzeuge/Port“ den virtuellen COM-Port aus, der GoNotify zugeteilt wurde.
- Programmierung:
  ATmega328PB: Wählen Sie das Board „GoNotify @ 4 MHz (internal RC)“. Wenn Sie auf die Schaltfläche „Hochladen“ klicken, startet die Programmierung.
  ESP-01S: Wählen Sie „Generic ESP8266 Module“. Drücken Sie die Trigger-Taste SW2 und halten Sie sie gedrückt, bevor Sie auf die Schaltfläche „Hochladen“ der IDE klicken. Hat die Programmierung begonnen, kann der Taster losgelassen werden.
- Öffnen Sie den „Seriellen Monitor“ zum Debuggen.

Wenn neue Firmware in das Gerät programmiert und der gesamte Speicher gelöscht wurde, muss dieser (re-)konfiguriert werden. Die Konfigurationsdaten sind im EEPROM der MCU gespeichert, die WLAN-Konfigurationsdaten im ESP-01S-Modul. Drücken Sie die Reset-Taste SW1. GoNotify arbeitet nun, wie in **Bild 4** zu sehen, als Access Point (AP) auf 192.168.4.1 (dies lässt sich in der Software ändern). Nach der Konfiguration erwartet GoNotify einen Trigger: Entweder drückt man SW2, schließt RE1 oder ein Sensor sorgt durch Über/Unterschreiten eines (zuvor in der Software definierten) Schwellwerts dafür. Der Summer erzeugt einen Meldungston und eine Testnachricht wird gesendet, deren Inhalt und Zielort davon abhängen, wie Sie das Gerät konfiguriert haben. Das Beispiel in Bild 4 sendet eine Nachricht an einen PushBullet-Dienst (www.pushbullet.com). Von diesem Punkt an arbeitet GoNotify im normalen Betrieb in seinem Low-Power-Guardmodus.

### Virtuelle Maschine

Nun, da Sie wissen, wie GoNotify funktioniert, wie man das Sensor-Interface baut und wie man es programmiert, können wir zu einigen praktischen Anwendungen übergehen. All die verkürzten URLs im Folgenden führen zu meinem GitHub-Repository [3]. Ich habe auch eine virtuelle Maschine (VM) mit allen Tools vorbereitet, damit Sie schnell loslegen können. Sie liegt im Ordner „GoNotify-Development-Env“ bei [3].

### Kalifornischer Wasserzähler mit Alarm in der Cloud

In diesem Beispiel überwacht GoNotify einen Wasserzähler über einen daran befestigten Reed-Schalter, der pro 500 ml Wasser einen Impuls erzeugt. Wenn ein Leck erkannt wird oder ein Wasserhahn zu lange offen ist, sendet GoNotify eine Benachrichtigung. Es meldet auch jede Stunde den Wasserverbrauch, damit man informiert ist, wann das meiste Wasser verwendet wird. Um Energie zu sparen, wird das ESP-01S-Modul nur eingeschaltet, um eine Warnmeldung zu senden, wenn es ein Problem gibt. Im Beispiel wird eine Warnung gesendet, wenn über einen Zeitraum von 24 Stunden keine 2-Stunden-Periode ermittelt wird, in der der gesamte Wasserverbrauch geringer als 0,5 l ist. Mit anderen Worten, bei einem kontinuierlichen Wasserverbrauch geht GoNotify davon aus, dass es ein Leck gibt. Auch wenn der Wasserverbrauch für 30 min oder länger hoch ist, ist GoNotify der Meinung, dass ein Wasserhahn offen gelassen wurde und sendet eine Warnung.
Wenn es mit der Internetverbindung ein Problem geben sollte, warnt GoNotify den Benutzer durch ein „lokales“ Summen. Für diese Anwendung muss der integ-

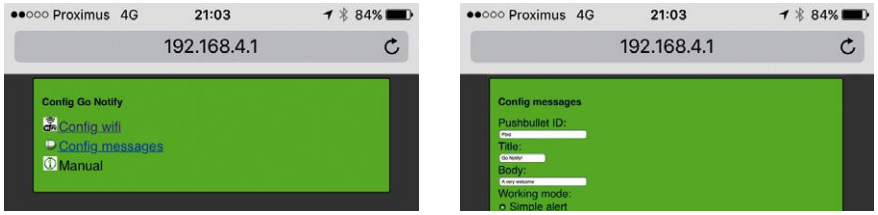



Bild 4. Verbinden Sie sich mit dem GoNotify-Accesspoint, um die Schaltung für Ihr WLAN zu konfigurieren.

rierte Temperatursensor des Mikrocontrollers kalibriert werden. Halten Sie die Triggertaste SW2 gedrückt (am besten mit einem angelöteten Kabelstückchen) und setzen Sie den Controller mit SW1 zurück. Kühlen Sie den Controller auf 0 °C herunter. GoNotify piept, wenn sich die Temperatur ändert, und schweigt, wenn die Referenztemperatur erreicht ist. Sie können nun SW2 loslassen beziehungsweise die Verbindung auftrennen.

- URL: https://wma-gonotify.rhcloud.com
- User: demo
- Passwort: demo
- Quellcode: https://goo.gl/UIkaXJ

In diesem Beispiel wird der ATmega328PB mit der Arduino-IDE programmiert, das ESP-01S-Modul mit dem nativen IoT-SDK2.0 (auf der Espressif-Website). Die Cloud-Lösung wird mit OpenShift von Redhat (www.openshift.com) erstellt und ist mit Nodejs und Mongodb programmiert. Es gibt mehrere Vorteile, wenn eine private Cloud genutzt wird:

- Sie allein besitzen die wertvollen und privaten Daten.
- Benachrichtigungen: GoNotify ist intelligent genug zu erkennen, dass ein Problem aufgetreten ist und eine entsprechende Warnung erstellt werden muss. Diese Warnungen werden der Cloud gemeldet, wo sie an ein Benachrichtigungssystem weitergeleitet werden. Auch wenn diese Warnung Sie nicht direkt erreicht, so wird das Problem (der Ausfall eines Sensors) in der Cloud gespeichert.
- Zeitsynchronisierte Messwerte: Die Cloud kann als Uhr für den Sensor fungieren. Bei jedem gemeldeten Messwert schreibt die Cloud in den REST-Header, wann die nächste Auslesung erwartet wird.

### MQTT-Client mit AllThingsTalk

Es ist sehr vorteilhaft, mit Arduino und mit beliebten Mikrocontrollern wie dem ATmega328PB und dem ESP8266 kompatibel zu sein, zumal auch viele IoT-Cloud-Plattformen eine ebenfalls kompatible API besitzen. Diese Anwendung zeigt, wie man mit nur einigen Programmzeilen GoNotify an die AllThingsTalk-Maker-API (https://maker.

## STÜCKLISTE

**Widerstände**
alle 1206
R1,R5,R11 = 1 k
R2,R7,R8,R12 = 47 k
R3 = 976 k*
R4 = 562 k*
R6 = 56 Ω
R9,R10 = 4k7
R13 = 100 k
R15,R16 = 27 Ω

**Kondensatoren:**
C1,C3,C5 = 10 µ, 16V, Tantal, 1206
C2 = 100 µ, 6V3, Tantal, 1206
C4,C7,C10,C11 = 100 n, 0805
C6 = 1 µ, 16 V, Tantal, 1206
C8,C9 = 47 p, 0805

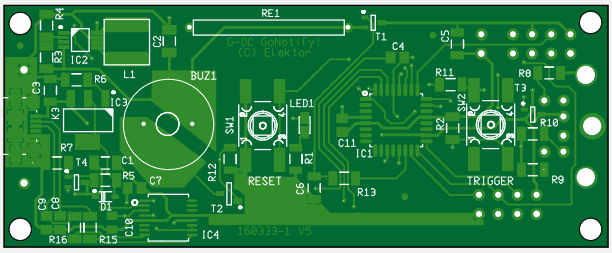
**Induktivität:**
L1 = 4µ7, $I_{rms}$=1A72, 0Ω082

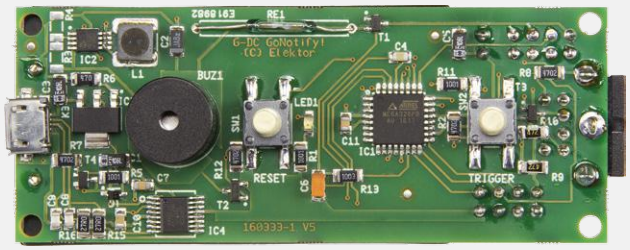
**Halbleiter:**
D1 = BAT760
LED1 = grün, 2012
T1,T3,T4 = NTS2101PT1G
T2 = 2N7002
IC1 = ATmega328PB-AU
IC2 = MCP1642**D**-ADJ*
IC3 = NCP1117ST33T3G
IC4 = FT230XS

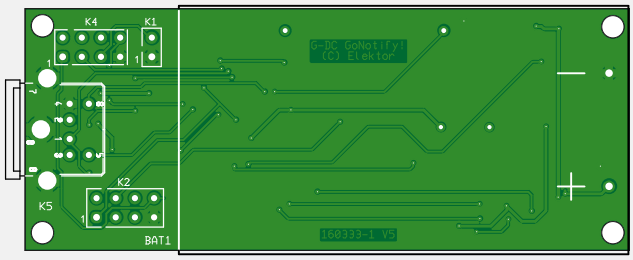
**Außerdem:**
BUZ1 = Summer, 12 mm Durchmesser, Raster 6,5 mm
K1 = 1x2-polige Stiftleiste, Raster 0,1"
K2 = 2x4-polige Stiftleiste, Raster 0,1"
K3 = Micro-USB-Buchse Typ AB
K4 = 2x4-polige Buchsenleiste, Raster 0,1"
K5 = 6-polige Mini-DIN-Buchse oder 4-poliger Grove-Verbinder*
RE1 = Reedschalter, Länge der Glasröhre 15 mm max.
SW1,SW2 = Tastschalter, SMD, 6,2 x 6,2 mm²

Batteriehalter, 2x AA (Keystone 2462)
WLAN-Modul ESP-01S ESP8266
Platine 160333-1

* siehe Text



allthingstalk.com) als MQTT-Client bindet. Das Beispiel veranschaulicht die Zwei-Wege-Kommunikation zwischen dem AllThingsTalk-Server und einem GoNotify-Klienten, gestattet die Fernsteuerung von GoNotify und gleichzeitig den Empfang von Statusinformationen. GoNotify ist in diesem Beispiel immer eingeschaltet und verbunden, muss also über USB versorgt werden. Der ATmega328PB wird mit der Arduino-IDE, das ESP-01S Modul mit dem nativen IoT-SDK2.0 programmiert.

URL: https://maker.allthingstalk.com/device/Kx7vo0kSYI5P6e9PrJIOKWS7
Zuerst muss ein eigenes AllThingsTalk-Konto eingerichtet, dann kann die Verbindung hergestellt werden.
Quellcode: https://goo.gl/OIF6Rw

## Überwachen eines Gewächshauses mit ESP-Now

ESP-Now ist ein proprietäres Kommunikationsprotokoll, das von Espressif entwickelt wurde. Es ermöglicht Geräten, Peer-to-Peer miteinander zu kommunizieren und dabei auf WLAN oder Handshaking zu verzichten. Es ist für entfernte, mit anderen mit ESP8266/ESP-01S verbundenen Sensoren gedacht, die mit dem Internet in Kontakt stehen (**Bild 5**).
In dieser Anwendung fungiert ein GoNotify als Brücke zum Internet und als sicherer MQTT-Client zum AllThingsTalk-Server, während ein anderer den Sensor ausliest. Die Brücke ist ein ESP-Now-Slave und bleibt immer eingeschaltet. Um sicherzustellen, dass der MQTT-Client aktiv bleibt (persists), werden periodisch Pings an den AllThingsTalk-Server gesendet, um die Verbindung zu überprüfen.
Der Sensor-GoNotify übernimmt die Rolle des ESP-Now-Controllers. Wenn er zum Beispiel ein Ereignis oder eine Warnung übertragen will, wacht er auf und verbindet sich mit der MAC-Adresse der Brücke, um ein paar Bytes zu senden und zu verifizieren.
In diesem Beispiel wird ESP-Now zur Überwachung eines Gewächshauses verwendet. Der Sensor-GoNotify ist mit einem DHT12-Sensor ausgestattet, um die relative Feuchtigkeit der Luft und die Temperatur zu messen. Ein weiterer Sensor überprüft die Feuchtigkeit des Bodens, um herauszufinden, ob die Pflanzen Wasser benötigen (**Bild 6**). Versorgt von Lithium-Batterien kann dieser GoNotify 400.000 Messwerte senden, sogar mit gewöhnlichen preiswerten Alkalibatterien ist die Übertragung von etwa 200.000 Messwerten möglich.
Für diese Anwendung wurden beide GoNotifys mit Arduino-Sketches programmiert.

- Quellcode für Feuchtesensor: https://goo.gl/9mqLos
- Quellcode für ESP-Now: https://goo.gl/18Qhd5

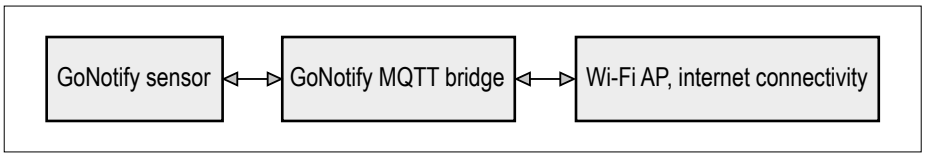



Bild 5. Das Kommunikationsprotokoll von ESP-Now ermöglicht Peer-to-Peer-Verbindungen zwischen mehreren ESP8266-Controllern.

Bild 6. GoNotify überwacht die Temperatur, die Luft- und die Bodenfeuchtigkeit in einem Gewächshaus.

## Von der Idee zum IoT

GoNotify kann als vernetztes Interface verwendet werden, um Daten von einem Sensor zu erfassen und sie online zu überwachen. Oder das Interface kann als einfaches ferngesteuertes System eingesetzt werden. Es ist Ihre Anwendung, die GoNotify in ein intelligentes Gerät verwandelt, das in der Lage ist, Warnungen zu erzeugen und Aktionen zu initiieren.
Am Ende ein Dankeschön an alle Open-Source-Enthusiasten. Viele Projekte stehen sowohl für die ESP8266- als auch für die ATmega328PB-Mikrocontroller open source zur Verfügung, ohne die die Entwicklung von GoNotify viel schwieriger geworden wäre.
Hut ab vor der Maker-Community!

(160333)

### Weblinks

[1] https://espressif.com/en/support/download/other-tools?keys=&field_type_tid%5B%5D=14

[2] www.elektormagazine.de/160333

[3] https://github.com/ginodecock/V3GoNotify/

[4] www.elektormagazine.com/labs/flexible-iot-sensor-interface-gonotify-1

### EINKAUFSZETTEL

→ 160333-1
GoNotify-Platine, unbestückt

→ 160333-41
Mikrocontroller ATmega328PB mit Bootloader

# Review:
# MiniPRO TL866A Programmer
## Viele Möglichkeiten für wenig Geld

Heutzutage sind ja die meisten Mikrocontroller mit einem Flash-Programmspeicher ausgestattet, der „in-circuit“ programmiert werden kann. Aber vor nicht allzu langer Zeit träumte jeder Hobby-Elektroniker von einem Gerät, mit dem er möglichst viele unterschiedliche (E)EPROMs, GALs oder Mikrocontroller programmieren konnte. Jetzt steht mit dem MiniPRO TL866A ein wahrer Alleskönner zu einem vernünftigen Preis zur Verfügung.

Von **Luc Lemmens**

Was soll das in unseren modernen Zeiten, in der die meisten programmierbaren Bauteile nur ein billiges USB-Interface zum Anschluss an den PC benötigen? Dann ist ein Programmiergerät doch kalter Kaffee! Oder doch nicht?
Es gibt genügend Gründe: Zum Beispiel Designs, bei denen ein ICSP-Anschluss auf der Platine hoffnungslos im Weg ist. Und es ist wahrscheinlich auch schon jedem passiert, dass der Anschluss bei einem Prototyp schlichtweg vergessen wurde. Auch lassen sich einige Einstellungen (Fuses) von Mikrocontrollern nicht über den ICSP-Anschluss ändern. Der MiniPro TL866A der chinesischen Firma XG Autoelectric [1] verfügt nicht nur über eine 2x20-polige Nullkraftfassung zur Aufnahme des zu programmierenden Bauteils, sondern besitzt auch selbst einen ICSP-Anschluss. Damit macht der Programmer ein Microchip-PICkit und/oder einen Atmel-AVRISP überflüssig. Der TL866

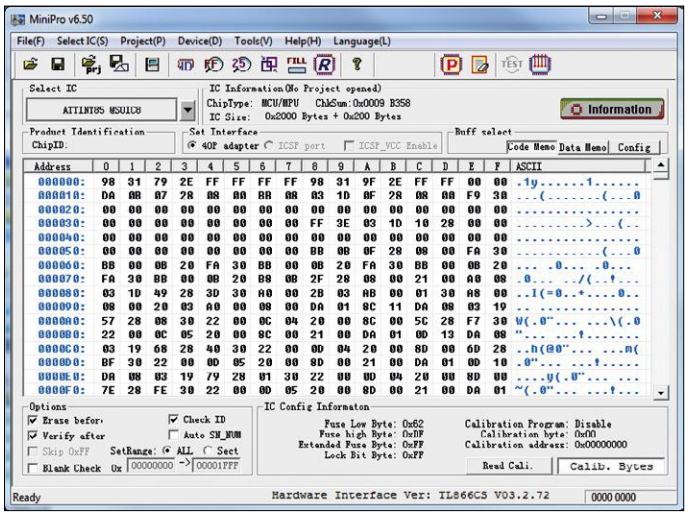



Bild 1. Die Windows-Software des MultiPro TL866A.

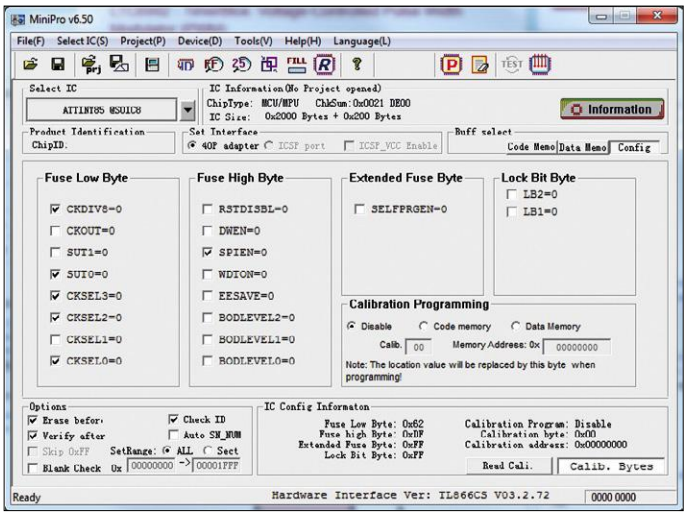



Bild 2. Die Einstellungen der Fuse-Bits.

unterstützt die Programmierung von über 14.000 verschiedenen Bauteilen, angefangen bei archaischen (E)EPROMs bis hin zu den neuesten Mikrocontrollern. Er kann sogar die Funktion von Logik-ICs (CMOS und TTL), DRAMs und SRAMs überprüfen.

## Software

Die mitgelieferte Windows-Software sieht ein wenig altmodisch aus, macht ihren Job aber richtig gut. Man muss nur manchmal etwas suchen, bis man die gewünschte Funktion gefunden hat. Es ist schön, dass auch „Projekte“ unterstützt werden: Dateien, die auch die Konfigurationsbits eines Mikrocontrollers zusätzlich zum Speicherinhalt enthalten.

Beim ersten Einsatz des TL866 stolperte ich zunächst über die Fuse-Einstellungen (Konfigurationsbits) des ATtiny85. Anscheinend gibt es oft Verwirrung, was es bedeutet, wenn ein bestimmtes Häkchen gesetzt ist: Ist die entsprechende Option nun aktiviert oder deaktiviert? Ist das Bit dann als „0“ oder „1“ programmiert? Bei einem Bit war es genau anders herum als in der Software des ELNEC-Programmers, den wir hier im Elektor-Labor verwenden. Glücklicherweise werden von der MiniPro-Software auch die endgültigen Konfigurationswörter in hexadezimaler Form angegeben, die genau den Angaben im Atmel-Datenblatt entsprechen. Im Zweifelsfall sollte man deshalb immer diesen kleinen Umweg gehen.

Auf der Wiki-Website der „MiniPro TL866 Community“ [2] findet man viele weitere Informationen für die Arbeit mit diesem Programmiergerät und sogar einen Treiber für Linux-Betriebssysteme.

## Ein Paket mit allem Zubehör

Wie gesagt, kann man vom TL866A aus Mikrocontroller von Atmel und Microchip auch über deren ICSP-Anschlüsse programmieren. Der mitgelieferte Adapter für Microchip-MCUs ist 1:1-kompatibel mit dem üblichen PICkit-Anschluss, für den 6- oder 10-poligen Atmel-AVR-Steckverbinder muss man selber ein Adapterkabelchen anfertigen.

Der Programmer mit seiner 2x20-poligen ZIF-Fassung wird über das USB-Kabel mit Strom versorgt. Die Version, die Sie jetzt im Elektor-Shop erwerben können, enthält noch sechs Adapter für ICs in PLCC- und in SO-Gehäusen, eine PLCC-Ausziehhilfe, eine ESD-geschützte Pinzette und ein ICSP-Kabel.

Vielleicht überflüssig, aber dennoch zur Sicherheit eine Warnung: Dieser Programmer ist für die Entwicklung von Prototypen oder Hobbyanwendungen prima geeignet. Aber die Programmieralgorithmen sind nicht zertifiziert. Zur Produktion kommerzieller Produkte sollte deshalb unbedingt ein zertifizierter professioneller Programmierer eingesetzt werden.

Davon abgesehen ist der TL866 auf jeden Fall ein tolles Gerät, vor allem, wenn man den Preis, die Funktionen und das mitgelieferte Zubehör berücksichtigt! ◀

(160355)

## Weblinks

[1] www.autoelectric.cn/en/TL866_main.html

[2] http://minipro.txt.si

Der TL866A unterstützt mehr als 14.000 verschiedene Bauteile.

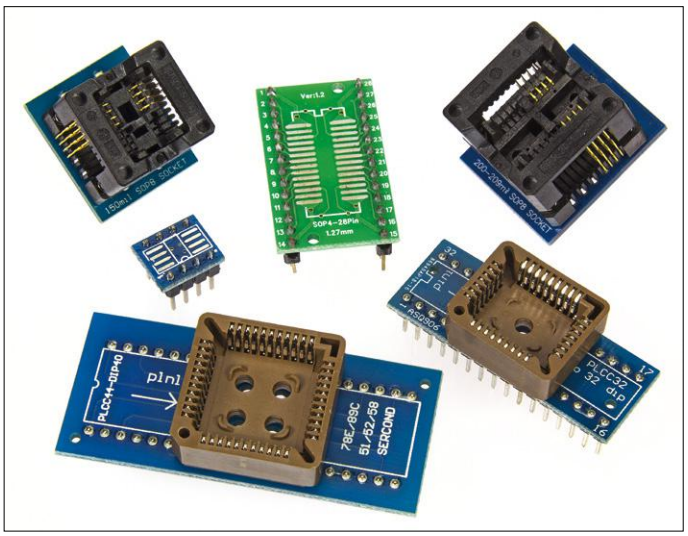
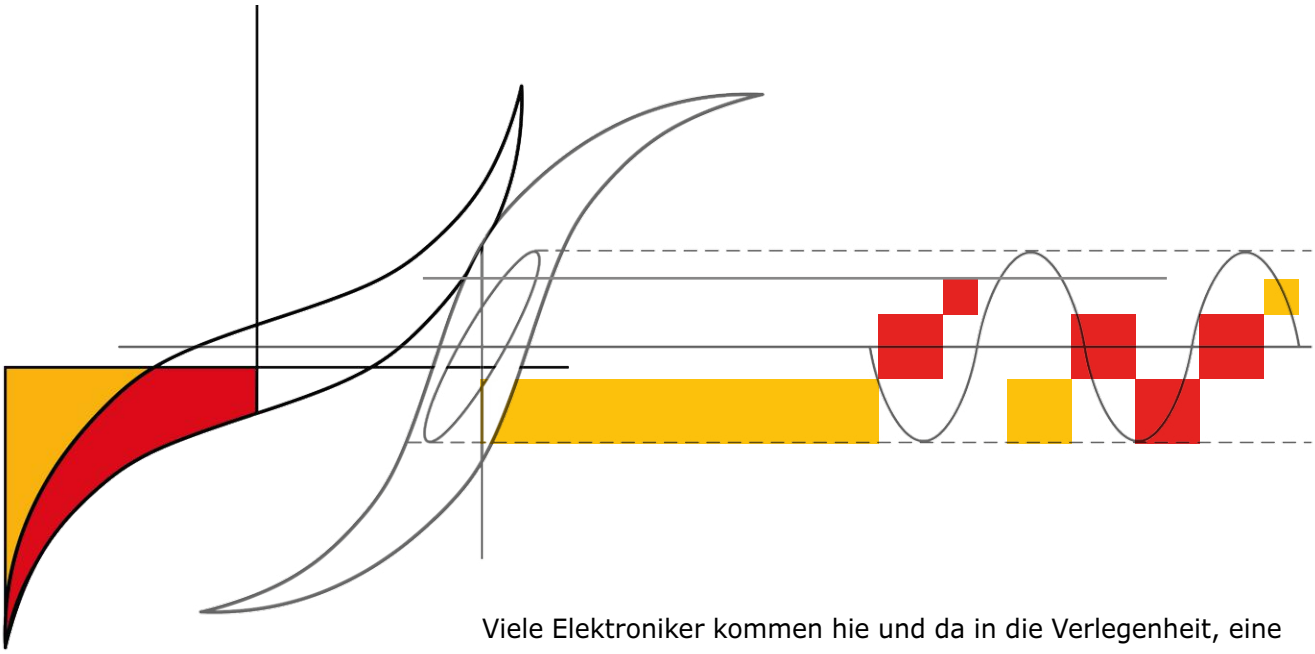



Bild 3. Sechs Adapter für ICs in PLCC- und SO-Gehäusen gehören zum Lieferumfang des Programmers.

# REDEXPERT: Simulator für induktive Bauelemente

## Spulen berechnen, auswählen und Verluste kalkulieren

Von **Ranjith Bramanpalli** (Würth Elektronik)



Viele Elektroniker kommen hie und da in die Verlegenheit, eine geschaltete Stromversorgung entwickeln zu müssen. Heute kein großes Problem mehr, denn für fast alle Zwecke gibt es passende ICs nebst Online-Design-Hilfen und Simulatoren der Chip-Hersteller. Doch gerade bei den induktiven Bauelementen, denen man aufgrund ihrer Komplexität fast mit Respekt begegnet, sieht es eher mau aus. Bisher. Denn jetzt gibt es das neue Online-Tool REDEXPERT von Würth Elektronik eiSos, dem Spezialisten für Induktives.

Leistungsverluste bei Schaltnetzteilen treten meist bei den Leistungshalbleitern und den induktiven Bauelementen auf. Verluste induktiver Bauelemente erfolgen im Kern und den Wicklungen. Die exakte Bestimmung dieser Verluste ist für zuverlässige Systeme mit hohem Wirkungsgrad wichtiger denn je. Zur Einschätzung der Kernverluste bei Schaltnetzteilen können komplexe Messaufbauten erforderlich sein, und doch lässt sich nicht garantieren, dass diese Abschätzungen für die spezifische Anwendung relevant sind. Kernverluste wurden anfangs mit der Steinmetzgleichung [1] bzw. mit deren Erweiterungen berechnet. Diese Gleichungen sind allerdings nur für bestimmte Bedingungen und Materialien zuverlässig. Daher wurde von Würth Elektronik eiSos ein neues Modell nach aktuellem Stand der Technik entwickelt, mit dem sich Kernverluste effektiv und genau ermitteln lassen. Dieses Modell wurde in im neuen Tool REDEXPERT [2] implementiert.

### Energiespeicherung bei Induktivitäten

In Schaltnetzteilen speichert die Speicherdrossel Energie in Form eines Magnetfeldes während der Einschaltphase des Schaltreglers und gibt sie während der Ausschaltphase an die Last ab. Die Speicherdrossel besteht üblicherweise aus einer Kupferdraht-Spule auf einem Kern mit bestimmten magnetischen Eigenschaften. Legt man eine magnetomotorische Kraft an eine Spule an, induziert diese im zeitlichen Verlauf einen Magnetfluss Φ(t). An jeder Stelle ist die Magnetflussdichte B stets proportional zur Feldstärke H.

$$B(t) = \mu_r \, \mu_o \, H(t)$$

Hier gilt: B = magnetische Flussdichte (Φ/A), $\mu_r$ = Material-Permeabilität, $\mu_o$ = Luft-Permeabilität und H = magnetische Feldstärke.

Die Spule ist um den Kern gewickelt oder darin untergebracht. Der Kern enthält einen Luftspalt, um das magnetische Feld effektiv zu nutzen. Der

Kern besteht in der Regel aus ferromagnetischem Ferritmaterial mit deutlich höherer Permeabilität als Luft. Aufgrund der Reihenschaltung des hohen Luftspaltwiderstands mit dem niedrigen Widerstand des Ferritmaterials wird der Großteil der Energie im Luftspalt gespeichert.

Spulen unterliegen den Gesetzen nach Ampère und Faraday. Das Ampèresche Gesetz setzt den Strom in der Spule zum Magnetfeld im Spulenkern ins Verhältnis. Zur Annäherung wird angenommen, dass das Magnetfeld im Kern über die gesamte Kernlänge ($l_e$) gleichförmig ist. Damit kann man das Ampèresche Gesetz wie folgt formulieren:

$$H \times l_e = N \times I$$

Wobei gilt: N = Anzahl der Windungen der Spule um den Kern und I= Strom durch die Spule.

Nach dem Faradayschen Gesetz beträgt die an der Spule anliegende Spannung:

$$U(t) = N \times \frac{d\emptyset}{dt} \text{ (bzw.) } U = L \times \frac{di}{dt}$$

Aus den obigen Gleichungen berechnet sich die Induktivität wie folgt:

$$L = \mu \frac{Ac\, n^2}{le}$$

Wobei gilt: Ac = Querschnittsfläche des Kerns.

Da Ferritmaterialien eine hohe Permeabilität und somit einen niedrigen magnetischen Widerstand aufweisen, stellen sie den einfachsten Weg für den Magnetfluss dar. Dies unterstützt den Fluss im Drosselkern, was Drosseln mit hohen Induktivitätswerten bei geringen Abmessungen ermöglicht. Dieser Vorteil ist aus der obigen Induktivitäts-Gleichung ersichtlich: Kernmaterial mit hoher Permeabilität ermöglicht eine kleinere Querschnittsfläche.

Bei Schaltnetzteilen lässt sich die maximale Magnetflussdichte wie folgt berechnen:

$$B_{pk} = \frac{L}{N \times Ac} \times \frac{di}{dt}$$

Auch gekoppelte Drosseln (wie etwa bei Sperrwandlern) sind Spulen, nur mit mehreren Wicklungen. Diese Wicklungen sind komplexer, aber die Kerneigenschaften bleiben davon unberührt.

## Leistungsverluste

Leistungsverluste treten sowohl in den Wicklungen als auch im Kern auf.

### Wicklungsverluste

Leistungsverluste in den Wicklungen entstehen aufgrund ihres Gleichstromwiderstands $R_{DC}$, sowie aufgrund des Skin-Effekts und des Proximity-Effekts. Leistungsverluste aufgrund von $R_{DC}$ lassen sich näherungsweise wie folgt bestimmen:

$$P = I^2 \times R_{DC}$$

Die durch den Skin- und den Proximity-Effekt bedingten Leistungsverluste werden als Wechselstromwiderstand $R_{AC}$ der Wicklung bezeichnet, der in erster Linie von der Frequenz abhängt. Es gibt einige Methoden zur Ermittlung dieser Effekte in magnetischen Bauteilen, doch sind zur näherungsweisen Bestimmung komplizierte Verfahren wie die Dowell-Methode nötig.

### Kernverluste

Zu deren Ermittlung wird die magnetische Flussdichte B gemessen, während die magnetische Feldstärke H erhöht wird. Das Verhältnis von B zu H ist nicht linear und zeigt eine Hysterese. Diese ist eine der typischen Eigenschaften des Kernmaterials, die zu Leistungsverlust im Spulenkern führen. **Bild 1** zeigt die typische BH-Kurve bei sinusförmiger Anregung eines Kerns.

Der Energieverlust eines Schaltzyklus im Kern entspricht der Differenz zwischen der Magnetenergie, die dem Kern während der Einschaltphase zugeführt wird, und der während der Ausschaltphase dank des Kerns zur Verfügung stehenden Energie. Die Elementarmagnete verursachen Verluste, da sie in der Ausschaltphase nicht von alleine in die Ausgangslage zurückfallen, sondern mit Energieaufwand zurückgestellt werden müssen.

Unter Verwendung der Gesetze von Ampère und Faraday lässt sich die Energie im Kern wie folgt ausdrücken:

$$E = \int H\, dB$$

Der Energieverlust im Kern ist der von der BH-Kurve umschlossene Bereich multipliziert mit dem Kernvolumen. Der Leistungsverlust entspricht dem Produkt aus E und der Schaltfrequenz.
Dieser Ausdruck ist unter den Bedingungen gültig, dass der Kern nicht in die Sättigung gefahren wird und die Schaltfrequenz im vorgesehenen (linearen) Betriebsbereich liegt. Der rote Bereich in Bild 1 stellt den Energieverlust dar. Je häufiger die Hysteresekurve durchlaufen wird, desto höher die Verluste. Insofern sind die Hystereseverluste direkt von der Frequenz abhängig.
Die zweite Form der Kernverluste entsteht durch Wirbelströme, die durch einen zeitlich veränderlichen Fluss

$$\frac{d\emptyset}{dt}$$

im Kernmaterial induziert werden. Nach der Lenzschen Regel wird durch eine Änderung eines Magnetflusses ein Strom induziert, der seinerseits einen Magnetfluss induziert, der dem ursprünglichen Fluss entgegenwirkt. Dieser Wirbelstrom fließt durch das leitfähige Kernmaterial und erzeugt nach $I^2$ x R Energieverluste.

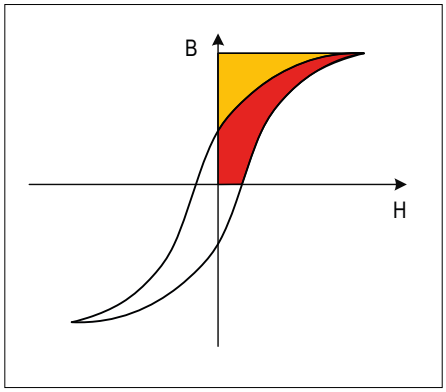



Bild 1. Typische BH-Kurve.

> Spulen simulieren

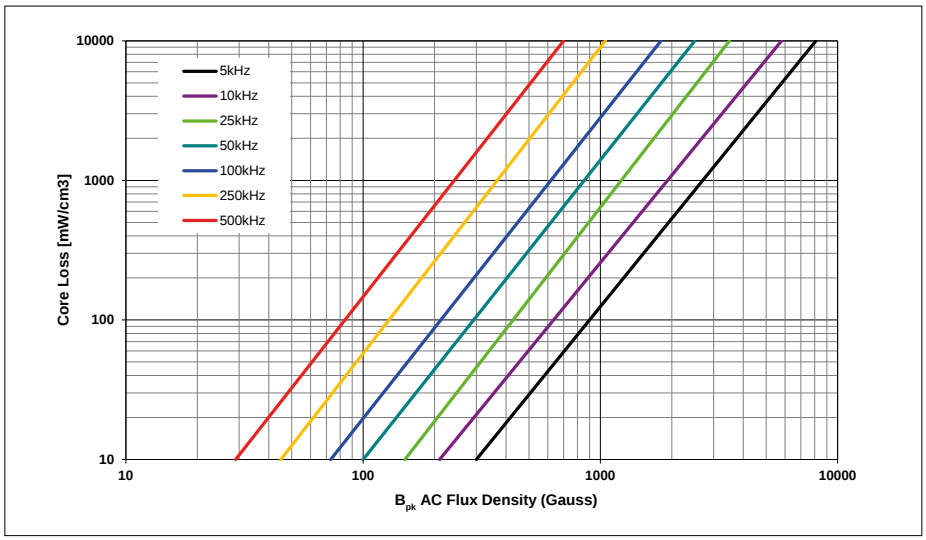



Bild 2. Kernverlustdiagramm bezogen auf die maximale Magnetflussdichte bei unterschiedlichen Frequenzen.

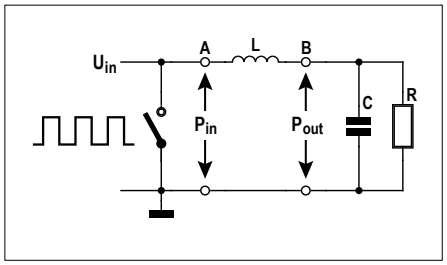



Bild 3. Prinzipaufbau des DC-DC-Wandlers zur Verlustbestimmung.

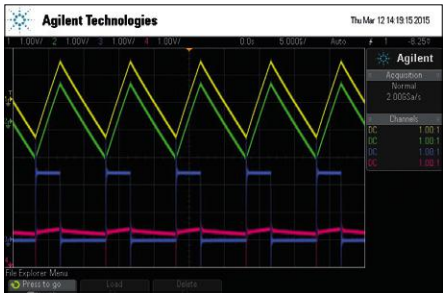



Bild 4. Screenshot des Oszillogramms der Schaltung von Bild 3.

## Ermittlung der Verluste

Anfänglich wurden Kernverluste mithilfe der als Steinmetz-Gleichung bekannten Leistungsgleichung ermittelt:

$$P_v = K \times f^{\alpha} \times B_{pk}^{\beta}$$

Wobei gilt: $P_v$ = Kernverluste (Hysterese und Wirbelströme) je Volumeneinheit, f = Frequenz, $B_{pk}$ = höchste Flussdichte bei sinusförmiger Anregung sowie K, $\alpha$ und $\beta$ = aus **Bild 2** abgeleitete Konstanten.

Die Kurven in Bild 2 zeigen die kombinierten Verluste der Hysterese und der Kernwirbelströme. Die Messung der Kernverluste ist schwierig, weil hierfür komplexe Aufbauten zur Messung der Flussdichte erforderlich sind, und zudem eine Schätzung der Fläche der Hysteresekurve notwendig ist. Zur Erstellung dieser Kurven wird eine Sinuswelle an den Ringkern mit ein oder zwei Wicklungen aus je einer Windung angelegt. Dann werden viele Daten generiert, um das Kernverlustdiagramm zu erstellen. Auf der Basis von Bild 2 werden dann die Konstanten abgeleitet.

Der wesentliche Nachteil der Steinmetz-Gleichung besteht darin, dass sie für sinusförmige Anregungen gilt. In Anwendungen der Leistungselektronik kommen aber vor allem nichtsinusförmige Magnetflussverläufe vor. Zwar gibt es auch Modelle, die das Problem nichtsinusförmiger Kurvenverläufe durch Trennung von Hysterese- und Wirbelstromverlusten zu lösen versuchen, doch hat sich die empirische Steinmetz-Gleichung als nützlichste Variante erwiesen. Sie bietet für sinusförmige Magnetflusswellenformen hohe Genauigkeiten bei einfacher Anwendung. Deshalb werden eher Erweiterungen genutzt, um sie auch für nichtsinusartige Magnetflusskurvenformen nutzbar zu machen.

Seit einiger Zeit ist die Erweiterung MSE (Modified Steinmetz Equation, [3]) gebräuchlich:

$$P_v = \left(K \times f_{eq}^{\alpha-1} \times B_{pk}^{\beta}\right) \times f$$

Hierbei gilt: $f_{eq} = \frac{f}{2\pi \times (DC - DC^2)}$

$f_{eq}$ = äquivalente Frequenz bezogen auf die Tastverhältnisänderung für nichtsinusartige Kurvenformen.
Aufgrund der mit MSE einhergehenden Nachteile wurde dann die GSE (Generalized Steinmetz Equation) entwickelt, die wir in nachfolgender Gleichung sehen.

$$P_v = \left(K \times f_{eq}^{\alpha} \times B_{eq}^{\beta}\right)$$

Hier gilt: $B_{eq} = \frac{1}{4}\int_0^T \left|\frac{dB}{dt}\right| dt$

Sowohl GSE- als auch MSE-Kernverlustdiagramme basieren auf einer sinusförmigen Erregung. Es gelten Einschränkungen, die weiter unten noch behandelt werden. Es gibt auch noch eine Reihe alternativer Modelle, die von den Herstellern optimal auf ihre Kerne abgestimmt wurden. Die **Nachteile der Steinmetz-Modelle** sind in einem extra Kasten aufgelistet.

## Das Wechselstromverlustmodell von Würth Elektronik

Würth Elektronik eiSos hat ein hochmodernes Modell entwickelt, mit dem man die passende Induktivität präzise auswählen und die Schaltung optimieren kann. Dieses Modell basiert auf den empirischen Daten, die aus Messungen an realen Aufbauten gewonnen wurden. Bei diesem Modell werden die Gesamtverluste der Induktivität in Wechselstromverluste (magnetische Aussteuerung der Spule und des Kerns) und Gleichstromverluste (Gleichstrom in der Wicklung) unterteilt.

Die Daten werden an einem DC-DC-Wandler wie in **Bild 3** erfasst. An die Induktivität wird eine pulsierende Spannung angelegt und dann die Eingangsleistung $P_{in}$ und die Ausgangsleistung $P_{out}$ gemessen. Auf dieser Basis wird $P_{Loss} = P_{in} - P_{out}$ geschätzt. Die Wechselstromverluste $P_{AC}$ der Spule werden separiert. Dieser Vorgang wird für unterschiedlichste Parametereinstellungen wie Schwankungen der magnetischen Aussteuerung, Frequenz, Rippelstrom usw. durchgeführt. Mithilfe der gewonnenen

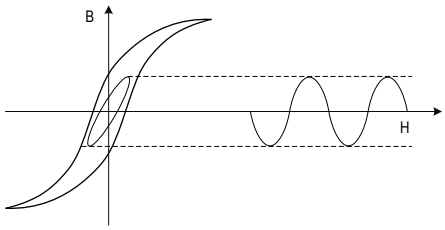



Bild 5. Umfassende Signalmodellierung für die Steinmetzgleichung und ihre Erweiterungen.

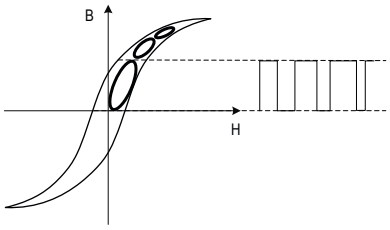



Bild 6. Signalmodellierung mit Hilfsschleifenmethode von Würth.

Daten wird das Modell zur Berechnung der Wechselstromverluste erstellt:

$$P_{AC} = f(\Delta I, freq, DC, k1, k2)$$

Die in den Datenblättern für typische Kernmaterialien gezeigte Hysteresekurven stellen die magnetische Aussteuerung durch ein sinusförmiges Signal von + nach – des Kerns dar. Die Hysteresekurve zeigt den in **Bild 5** dargestellten Energieverlust. Dieser Ansatz ist mit dem zur Erzeugung empirischer Daten für das in Bild 2 gezeigte Kernverlustdiagramm identisch.

Bei einer Schaltnetzteilanwendung wird jedoch der Kern durch ein wesentlich kleineres Rechtecksignal mit maximaler Flussdichte betrieben, die durch die Kernverluste auf eine kleinere Hysteresekurve beschränkt wird (**Bild 6**). Der Leistungsverlust hängt davon ab, wie häufig die Hystereseschleife durchlaufen wird. Insofern sind die Hystereseverluste auch direkt von der Frequenz abhängig. Die Hysteresekurve ändert ihre Form bei Änderungen der Wellenform, des Betriebsstroms bzw. der Betriebsspannung und der Temperatur. Diese Schwankungen machen eine genaue Vorhersage der Kernverluste ausgesprochen schwierig. Die kleinere Hysteresekurve hängt von der Spannung über der Drossel ab. Diese kleinere Hysteresekurve wird genau am Arbeitspunkt zur Erzeugung empirischer Daten für das Wechselstromverlustmodell von Würth verwendet. Dies hat sich als robust und präzise für weite Bereiche von Parametern wie Frequenz, Rippelstrom und Tastverhältnis erwiesen. Die **Vorteile des Verlustmodells von Würth Elektronik** sind in einem extra Kasten aufgeführt.

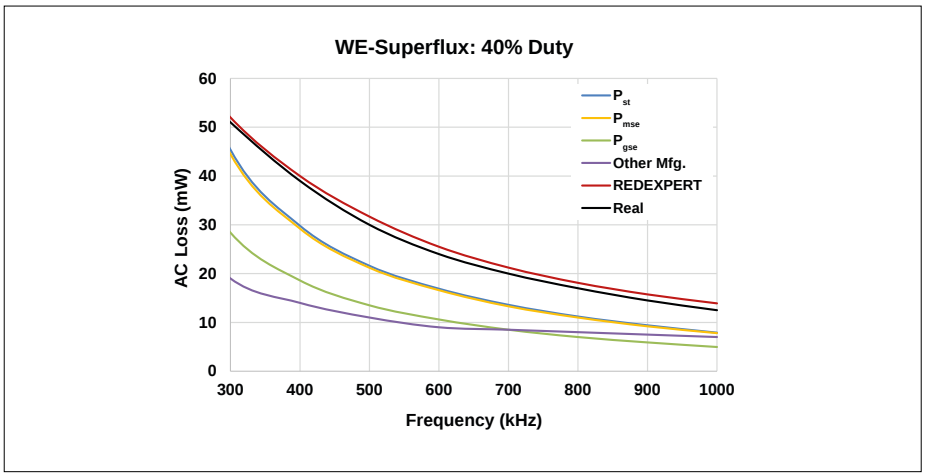



Bild 7. Kern aus WE-Superflux bei Duty Cycle von 40 %.

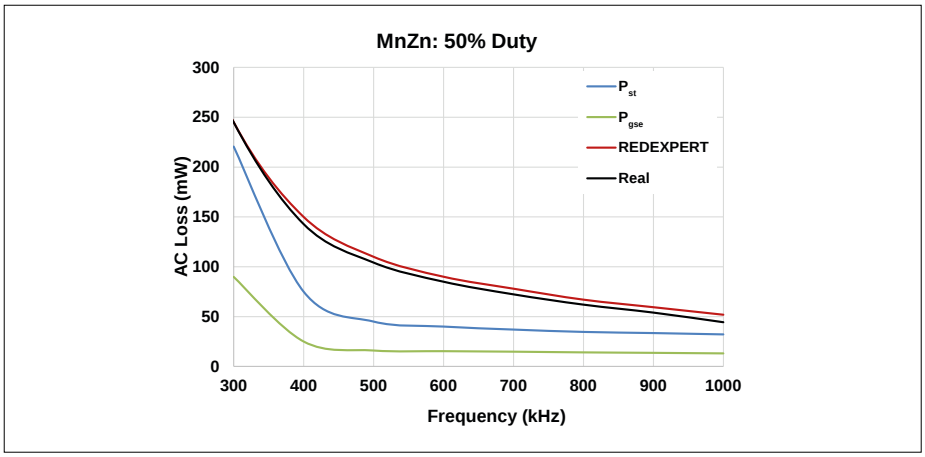



Bild 8. Kern aus MnZn bei Duty Cycle von 50 %.

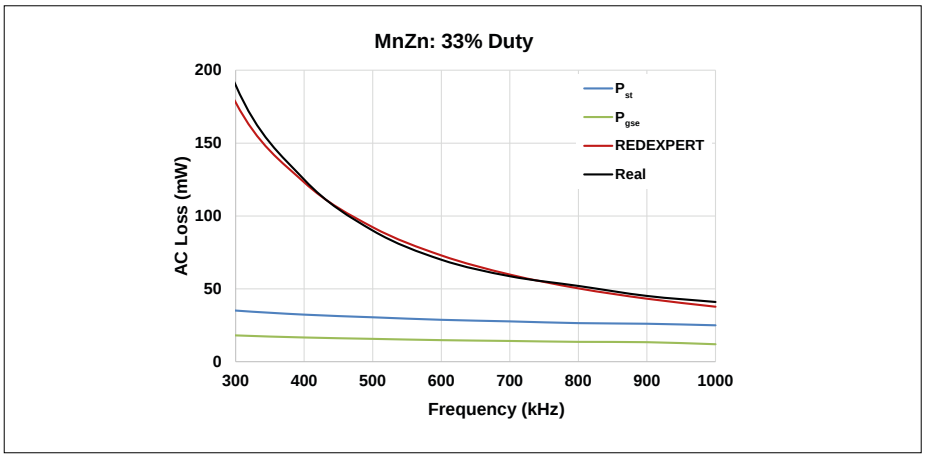



Bild 9. Kern aus MnZn bei Duty Cycle von 33 %.

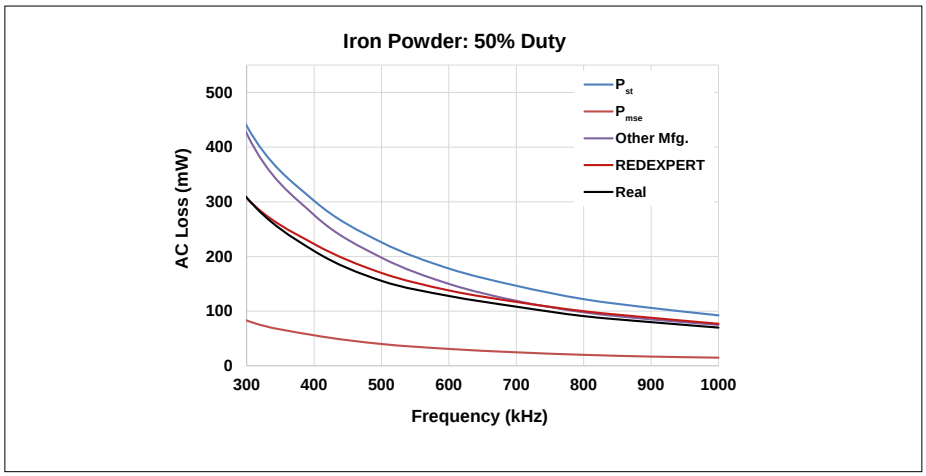



Bild 10. Kern aus Eisenpulver bei Duty Cycle von 50 %.

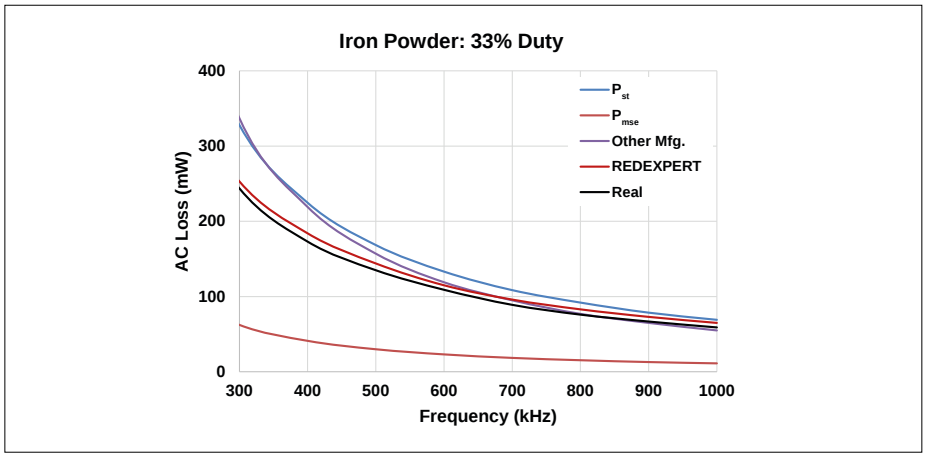



Bild 11. Kern aus Eisenpulver bei Duty Cycle von 33 %.

Bild 12. Screenshot des Online-Tools REDEXPERT.

Das Modell von Würth Elektronik wurde ausgiebig empirisch validiert und mit bestehenden Modellen und gemessenen Daten verglichen. Wechselstromverluste für verschiedene Materialien wie WE-Superflux, Eisenpulver, NiZn, MnZn usw. wurden über große Tastverhältnis- und Frequenzbereiche gemessen und mit theoretischen Modellen verglichen (**Bilder 7...11**). In diesen Diagrammen entsprechen die Kurven *Pst* der Steinmetzgleichung, *Pmse* der Modified Steinmetz Equation und *Pgse* der Generalized Steinmetz Equation. *REDEXPERT* ist der Wechselstromverlust laut dem AC-Verlustmodell von Würth und *Real* ist der gemessene AC-Verlust.

## Leistungsverluste mit REDEXPERT bestimmen

REDEXPERT ist das neue Online-Tool zur Auswahl und Simulation induktiver Bauelemente von Würth Elektronik, mit dem man unkompliziert eine geeignete Speicherdrossel auswählen kann. Es handelt sich um ein einfach zu bedienendes und effektives Tool (**Bild 12**), mit dem man Induktivitäten in kürzester Zeit vergleichen und auswählen kann. Zunächst gibt man die Eingangs- und Ausgangsparameter in der gewünschten Topologie ein. Dann berechnet REDEXPERT den Wert der Induktivität und zeigt passende Bauteile an.

Da Würth Elektronik eine umfangreiche Palette von Speicherdrosseln für alle möglichen Einsatzzwecke anbietet, sollte man für praktisch jede Anwendung nach Eingabe der Parameter ein geeignetes induktives Bauteil finden. Die kritische und normalerweise auch komplexe Berechnung der AC-Verluste in einem magnetischen Bauteil wird durch REDEXPERT dank des integrierten präzisen AC-Verlustmodells massiv vereinfacht. Aufgrund der genauen Berechnung der kompletten AC-Verluste eignet sich das Tool auch zur Temperaturabschätzung.

**Bild 13** zeigt das Beispiel eines Down-Konverters. Als Eingangsspannung werden 8...12 V angenommen und als Ausgangsspannung 5 V. Weitere Parameter sind Schaltfrequenz = 800 kHz, Rippelstrom = 40 % und Ausgangsstrom = 1 A. Für diese Werte errechnet REDEXPERT eine optimale Induktivität $L_{opt}$ von 9,6 µH, eine Einschaltdauer von 550 ns und ein Tastverhältnis von 0,44. Für dieses Beispiel werden nun mehr als 200 Produkte zur Auswahl vorgeschlagen. Will man eine sehr kleine und verlustarme Speicherdrossel haben, so empfiehlt sich ein Exemplar aus der WE-MAPI-Baureihe.

Gegenwärtig unterstützt REDEXPERT drei Topologien: Up-, Down- und SEPIC-Konverter. Außerdem gibt es einen Verlustrechner, der unabhängig von der Topologie die Verluste für Speicherdrosseln

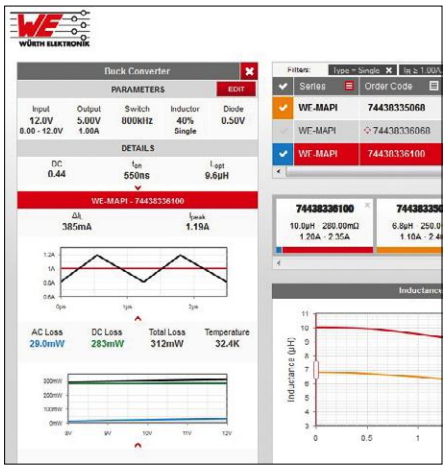



Bild 13. Beispiel mit Buck-Konverter.

berechnet. Da REDEXPERT webbasiert ist, muss man nichts downloaden oder installieren und hat jederzeit die neueste Version zur Verfügung.

## Fazit

Das Wechselstromverlustmodell von Würth Elektronik ist ein genaues und praxisgerechtes Modell zur Bestimmung von Wechselstromverlusten. Es wurde über weite Wertebereiche für die Parameter Frequenz, Rippelstrom und Tastverhältnis empirisch validiert und hat sich dabei als sehr robust erwiesen. Da dieses Wechselstromverlustmodell in REDEXPERT implementiert ist, erübrigt sich der Umgang mit Kernverlustdiagrammen. REDEXPERT ist ein Online-Tool zur Berechnung der induktiven Bauteile von Schaltreglern und somit eine ideale Ergänzung ähnlicher Tools von Herstellern von Schaltregler-ICs. ⏮

(160539

### Nachteile der Steinmetz-Modelle

- Abhängigkeit von empirischen Daten der Kernhersteller: Für Kernverlustdiagramme muss auf die Daten des Kernherstellers zurückgegriffen werden. Hersteller passiver Bauteile haben keinen Einfluss auf den Testaufbau.
- Geringe Genauigkeit bei Puls- und Dreieckswellenformen, da die Kernverlustdiagramme auf Daten basieren, die durch sinusförmige Anregung entstehen.
- Aufgrund von Fehlern bei der Parameterkonvertierung funktioniert die Erweiterung des Steinmetzmodells nur bei einem Tastverhältnis von 50 % und einem beschränkten Frequenzbereich optimal.
- Beschränkung auf Bauteile, die aus bestimmten Materialien bestehen oder von bestimmten Herstellern stammen.
- Wegen der hohen Komplexität bei der Schätzung der Magnetweglänge ist die Schätzung der Kernverluste mittels bestehender Modelle für Eisenpulver- und Metal-Alloy-Materialien nicht nur anspruchsvoll, sondern auch starken Schwankungen unterworfen.
- Aufgrund zeitlicher Magnetflussdichteänderungen entstehen auch in den Wicklungen Verluste infolge von Skin- und Proximity-Effekten. Die oben beschriebenen Ansätze berücksichtigen keine Wechselstromverluste der Wicklung.
- Bei Bauteilen aus mehreren Materialien ist eine Schätzung der Verluste nicht möglich.
- Hohe Komplexität von Aufbauten, um empirische Daten ausreichender Genauigkeit zu ermitteln.

### Vorteile des Verlustmodells von Würth Elektronik

- Da die empirischen Daten auf Echtzeitparametern basieren, lässt sich eine genaue Ermittlung der Verluste für jedes gegebene Tastverhältnis erzielen.
- Das Modell funktioniert über einen weiten Frequenzbereich (10 kHz bis 10 MHz) sehr genau, da die Konstanten der Leistungsgleichung unter Einbeziehung der magnetischen Aussteuerung über einen weiten Bereich ermittelt wurden.
- Auch geringfügige Änderungen im Kernmaterial und der Wicklungsstruktur werden berücksichtigt.
- Gültig auch für Bauteile aus mehr als einem Material.
- Gestattet die exakte Ermittlung von Verlusten bei Eisenpulver und neuen „Metal Alloy“ Materialien.
- Gültig für beliebige Kernformen und Windungsstrukturen.
- Beinhaltet AC-Wicklungsverluste.

### Weblinks & Literatur

[1] Steinmetzgleichung: http://de.wikipedia.org/wiki/Steinmetzformel
„On the law of hysteresis“ von Steinmetz, C.P.
„Improved calculation of core loss with nonsinusoidal waveforms“ von Jieli Li; Abdallah, T.; Sullivan, C.R.

[2] REDEXPERT: www.we-online.com/redexpert

[3] Erweiterte Steinmetzgleichung:
„Calculation of losses in ferro- and ferrimagnetic materials based on the modified Steinmetz equation“ von Reinert, J.; Brockmeyer, A.; De Doncker, R.W.

# Als die Bilder sprechen lernten

## Richtig „stumm“ war der Film eigentlich nie

Von **Peter Beil**

Wenn uns heutzutage im Kino akustische Effekte fast aus dem Sessel reißen, oder leisestes Flüstern dank modernster Tontechnik auch noch in der letzten Reihe zu verstehen ist, lohnt es sich, mal einen Blick zurück zu werfen, wie alles angefangen hat…

In der Frühzeit des Kinos sorgten Künstler am Klavier, der Kino-Orgel (**Bild 1**) oder ganze Orchester für die tonliche Untermalung. Allerdings beruhte das immer auf Interpretationen der Musiker und hatte kaum synchronen Bezug zum Bild.

Dabei hatte der bei T. A. Edison angestellte Chef-Ingenieur William Dickson schon 1893 den Phonographen mit dem Kinetographen zum „Kinetoscope“ kombiniert (**Bild 2**) und auf der Weltausstellung in Chicago präsentiert. Das Ganze funktionierte aber nur mit kurzen Filmstreifen oder Filmschleifen.

**Kratzig, aber einigermaßen synchron: Der Nadelton**

Was lag also näher, als den inzwischen technisch verbesserten Kinoprojektor mit einem Grammophon zu verbinden? Schon 1901 ließ sich die französische Firma Gaumont diese Technik patentieren. Der erste bekannte synchrone Bezug einer Musik zum Bild war 1908 eine „maßgeschneiderte“ Komposition von Camille Saint-Saens, zum Film „L‘Assassinat du Duc de Guise“, verkoppelt mit einer Schallplatte.

Der eigentliche Durchbruch dieses Verfahrens kam aber erst 1924 als Warner Bros. Entertainment das System als „Vitaphone“ auf den Markt brachte (**Bild 3**). Die dabei verwendete Schallplatte hatte einen Durchmesser von 17 Zoll (ca. 43 cm) und erreichte mit 33 1/3 Umdrehungen die Laufzeit einer damals üblichen Filmrolle von 300 Metern. Zum synchronen Start durch den Vorführer hatten Filmrolle und Platte eine Startmarke. Wie bei der heutigen CD wurde die Platte von innen nach außen abgetastet. Der bekannteste Film nach diesem Verfahren hatte 1927 Premiere und zwar „The Jazz Singer“, der Al Jolson weltberühmt machte (**Bild 4**).

Bild 1. Die Wurlitzer Kino-Orgel.

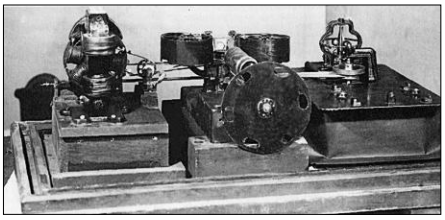


Bild 2. Dickson‘s Kinetoscope.

Bild 3. Vitaphone Projektor.

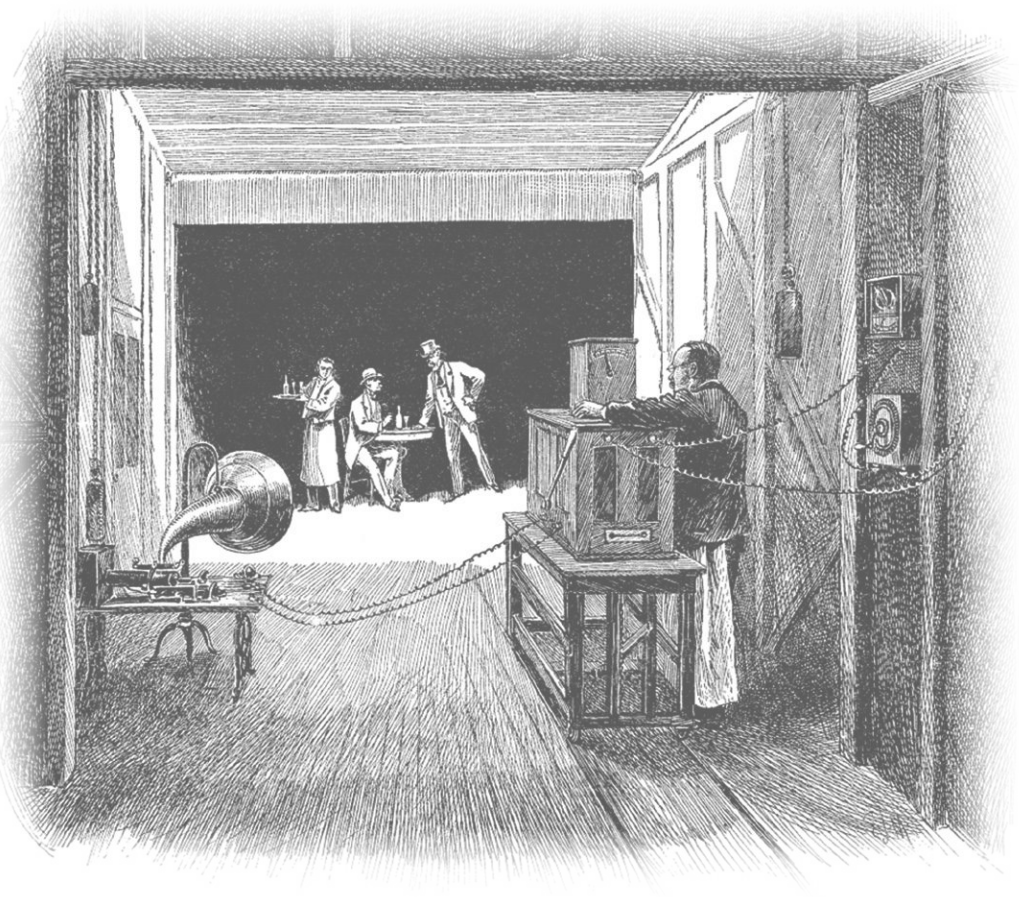


Bild 4. Premiere von „The Jazz Singer“.

Bild 5. Links Stummfilm, rechts Tonfilm.

## Aus Licht wird Ton

Das über Jahrzehnte weltweit eingesetzte Tonverfahren war aber der Lichtton. Erst damit konnten Bild und Ton, auf einem gemeinsamen Medium, synchron vereint werden. Der Lichtton hatte in den zwanziger Jahren mehrere „Väter“. Neben dem polnischen Ingenieur Józef Tykocinski-Tycociner, den deutschen Ingenieuren Hans Vogt, Joseph Massolle und Benedikt Engl, zählt auch der Schwede Sven Berglund zu den Pionieren des Lichttons. Als dann der Amerikaner Lee de Forest ebenfalls ein Patent anmeldete, war das der Auftakt zu einem jahrelangen Patentstreit, der erst 1930 im sogenannten „Tonfilmfrieden“ beigelegt wurde. Damit gab es ein weltweit einheitliches System.

## Endlich lippensynchron — mit Lichtton

Die Lichttonspur hat eine Breite von 2,19 mm und wurde durch Verkleinern der Filmbildfläche zwischen Perforation und Bild untergebracht (**Bild 5**). Dabei unterscheidet man zwei grundsätzliche Verfahren: Die Sprossenschrift (*variable density*), bei der die Modulation durch unterschiedlich starke Belichtung bei konstanter Spurbreite erfolgt (**Bild 6B**) und die Zackenschrift (variable area), wo das durch unterschiedliche Breite der Belichtung geschieht (**Bild 6A, -C**). Eine weniger verbreitete Version war die „Multi-Zacke“ (**Bild 6E**). Da die Wiedergabequalität der Sprossenschrift maßgeblich von der sauberen Entwicklung des Negativs und der Kopie abhing, verwendete man hauptsächlich die sogenannte „Doppelzacke“, da sie auch kopiertechnisch einfacher zu verarbeiten war. Leider geschah es anfangs häufig, dass die feinen Spitzen der Zacken durch zu langes Entwickeln verschmolzen, was einen Verlust der hohen Töne bedeutete. Um das durch die weißen Flächen entstehende Rauschen zu reduzieren, hat man in späteren Jahren, bei der „Eurocordschrift“, die Weißflächen durch eine dynamische Schwarzblende zugedeckt (**Bild 6D**). Das musste in gleichmäßig weicher Kurve erfolgen, damit das bei der Wiedergabe nicht als „Ton“ interpretiert wurde.

## So „liest“ man Licht

Ausgelesen wird die weltweit genormte Lichttonspur über einen 0,018 mm schmalen Spalt. Im Prinzip bedeutet eine schmale Spur einen leisen Ton (**Bild 7A**), eine breite Spur einen lauten (**Bild 7D**). Langsame Spurbreitenänderungen bewir-

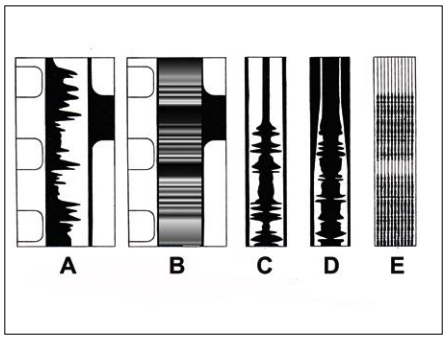



Bild 6. Die sechs Lichttonsysteme. A: Einfachzacke; B: Sprossenschrift, C: Doppelzacke; D: Eurocordschrift; E: Multizacke.

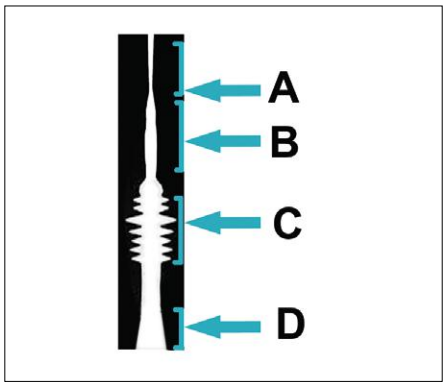



Bild 7. Definition der Spurbreite. A: leiser Ton; D: lauter Ton; B: tiefer Ton; C: hoher Ton.

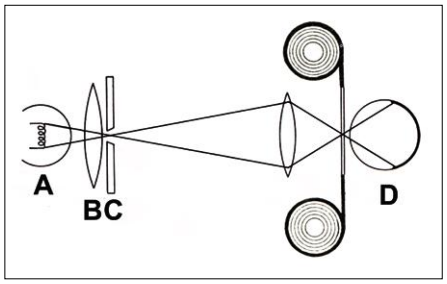



Bild 8. Das Prinzip des Lichttons. A: Tonlampe; B: Kondensor; C: Spalt, D: Fotozelle.

**Vorgetäuscht!**

Die Filmwiedergabe beruht auf der Trägheit des menschlichen Auges. Es werden in schneller Folge einzelne zeitversetzte Standbilder gezeigt. Das geschieht im Kinoprojektor mit dem sogenannten Malteserkreuz-Getriebe (**Bild 15**). Segment A dreht sich kontinuierlich und transportiert das Malteserkreuz B mit dem Zapfen D jeweils eine Phase weiter. Im Stillstand verriegelt das Segment C den Transport. Das Bild steht somit eine 24stel Sekunde, dann deckt eine Blende das Bild ab und in der Zeit wird der Film ein Feld weiter transportiert und das nächste Bild gezeigt. Um den dabei entstehenden Flacker-Effekt möglichst klein zu halten, hat man sich auf eine Bildfrequenz von 24 B/Sek. für das Kino geeinigt.

Bild 9. Tonlampe.

Bild 10. Fotozelle.

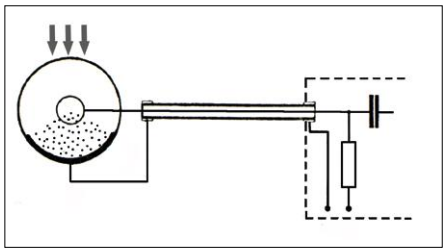


Bild 11. Elektrisches Prinzip der Fotozelle.

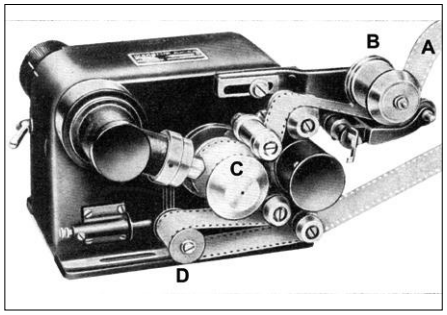



Bild 12. Europa-Lichttongerät. A: Filmschleife; B: Bremsrolle; C: Schwungmasse; D: Spannrolle.

ken einen tiefen Ton (**Bild 7B**), schnelle ergeben hohe Frequenzen (**Bild 7D**). Der Weg des Films beträgt bei 24 Bildern in der Sekunde 456 mm. Bei der genormten Spaltbreite von 0,018 mm erreicht man eine theoretische Abtastmöglichkeit von

456 / 0,018 = 25361

halben Schwingungen/Sek., also rund 12.500 Hz. Bei optimalen Verhältnissen wurden dabei immerhin 40 dB Störabstand erreicht. Das mag unter heutigen Gesichtspunkten eher „dürftig“ erscheinen, aber in Verbindung mit den damals üblichen Kinolautsprechern, mit riesigen hölzernen Schallwänden, Druckkammerhörnern und Röhrenverstärkern (mit harmonischen Klirrfaktoren), ein durchaus befriedigender Klang!

## Die Technik

Die Wiedergabe basiert auf einem ausgeklügelten System aus Optik, Elektronik und Mechanik (**Bild 8**). Die sogenannte „Tonlampe“ (A) bündelt durch einen Kondensor (B) das Licht auf den oben beschriebenen Spalt (C), der absolut senkrecht zur Tonspur stehen muss, da sonst non-lineare Verzerrungen entstehen. Der erzeugte Lichtstreifen wird mit einer Optik präzise auf die Lichttonspur projiziert. Auf der anderen Seite des Films trifft diese Information auf die Fotozelle (D), welche die Lichtimpulse wieder in elektrische Informationen umwandelt, die ein entsprechender Verstärker als Töne hörbar macht.

Besagte Tonlampe ist ein Spezialmodell mit besonders stabilem Wendel (**Bild 9**). Da ein großer Kinoprojektor immer etwas vibriert, würde sonst ein „Klingeln“, durch Zittern der Glühdrähte, die Modulation überlagern. Auch muss die Lampe mit Gleichspannung betrieben werden, um kein Netzbrummen zu erzeugen. Die Sockel haben eine Kerbe, damit man bei Ausfall die Lampe diese in Sekundenschnelle austauschen kann. Die Fotozelle arbeitet nach dem Prinzip der Elektronenröhre und ist wegen Ihrer Lichtempfindlichkeit mit rotbraunem Schutzlack überzogen (**Bild 10**). Charakteristisch ist die ringförmige Anode, an der die positive Spannung liegt. Der negative Pol ist die dahinter liegende lichtempfindliche Schicht, die Kathode. Sobald die Lichtbestrahlung einsetzt, wandern freiwerdende Elektronen von der Kathode zur Anode: es fließt ein elektrischer Strom, der am Eingangswiderstand des Verstärkers als Wechselspannung abfällt (**Bild 11**). Die Verbindung zum Verstärker erfolgt über ein kapazitätsarmes Kabel, um keine hohen Frequenzen zu verlieren. Die Fotozelle wurde natürlich mehrfach weiter entwickelt und später durch die Fotodiode ersetzt.

## Aus „Ton“ wird „Licht“

Die Herstellung der Lichttonspur erfolgte im Prinzip auf die gleiche Weise. Bei der Sprossenschrift wurde die Helligkeit der Lichtquelle von der Modulation gesteuert. Bei der Zackenschrift bewegten sich, abhängig vom Tonpegel, eine bzw. zwei Lamellenblenden vor dem Spalt, woraus sich die Zackenschrift ergab. Belichtet wurde auf besonders kontrastreich arbeitendem Filmmaterial, das so gut wie keine Grautöne erzeugte.

Beim Tonfilm ist die Lichttonspur dem Bild um 20 Felder voraus. Würde man nämlich den Ton im Bildfenster abtasten, wäre das Ergebnis ein lautes Knattern. Der Filmtransport hat ja eine Transport- und eine Standphase (siehe Kasten). Zur Tonabtastung muss diese ruckartige Bewegung also wieder geglättet werden. Dies geschieht im „Tongerät“ (**Bild 12**) zuerst mit einer Filmschleife (A) und einer mit Friktion ausgestatteten Bremsrolle (B). Danach folgt eine

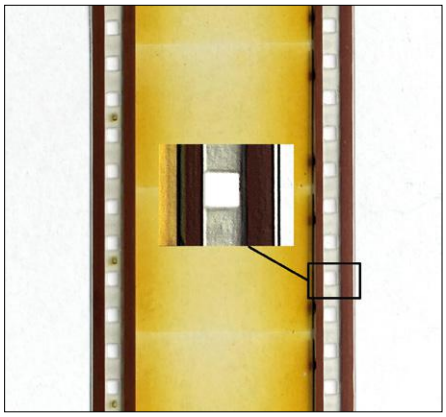


Bild 13. 4-Kanal Magnettonfilm.

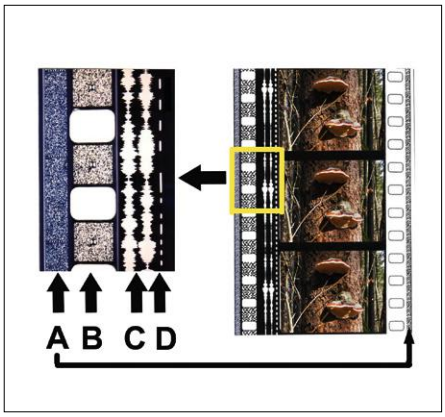



Bild 14. Anordnung der modernen Lichtton-Spuren. A: Sony SDDS; B: Dolby SR; C: Dolby Stereo; D: Timecode für DTS.

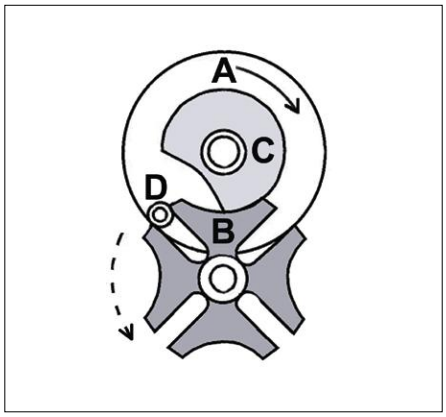



Bild 15. Malteserkreuz-Getriebe. A: Antrieb; B: Malteserkreuz auf der Transportachse; C: Sperrscheibe für Standphase; D: Transportzapfen.

Schwungmasse (C) und eine federnde Spannrolle (D), diese gleicht letzte Unebenheiten aus.

### Der Ton wird „bunt“

Mit dem Einzug des Farbfilms kamen ein paar neue Probleme. Bis dahin hatte man sich keine Gedanken über die Farbempfindlichkeit einer Fotozelle gemacht. Um auch auf dem Farbmaterial eine kontrastreiche schwarz-weiße Lichttonspur zu bekommen, waren jetzt einige aufwändige kopiertechnische Verfahren nötig. Um das zu vereinfachen, hatte man die Cyan-farbene Tonspur entwickelt, die aber mit rotem Laserlicht abgetastet werden musste. Da die Umrüstung der Theater nicht schnell genug voran kam, führte man für die Übergangszeit eine Magenta-farbene Tonspur ein, die mit Weiß- und Rotlicht gleichermaßen funktionierte.

### Das Magnetton-Intermezzo

Anfang der fünfziger Jahre erreichte der Magnetton die Filmtontechnik. Dazu wurden beim 35 mm Film vier Tonspuren auf den Film aufgetragen, für Links, Mitte, Rechts und einen Effektkanal (**Bild 13**). Letzterer wurde, zur Unterdrückung des Leerrauschens, durch ein Gate-Filter nur bei Schallereignissen zugeschaltet. Da der 35-mm-Film eigentlich keinen Platz für weitere Tonspuren bot, waren die Perforationslöcher etwas schmaler, um auch die Filmränder nutzen zu können.
Aus Platzgründen war das Magnetton-Wiedergabegerät oberhalb des Bildfensters angebaut, der Ton war damit um 28 Felder zurück versetzt.
Ferner musste der Projektor immer wieder entmagnetisiert werden, die Zahnrollen waren wegen der kleineren Perforationslöcher zu wechseln und die Filmspulen waren gegen solche aus Kunststoff auszutauschen. Dazu war der Magnetkopf einer recht hohen Abnutzung unterworfen. Wegen der aufwändigen Handhabung und der hohen Kosten für die Umrüstung der Kinos verschwand der Vier-Kanal-Magnetton bald wieder in der Versenkung.

### Die Erweckung des Lichttons aus dem Dornröschenschlaf

Nachdem der Lichtton runde fünfzig Jahre auf gleichem technischen Niveau verharrte, erbarmte sich die Firma Dolby des Systems. „Dolby A“ verbesserte ab 1976 die Tonqualität erheblich und es konnten nun zwei Tonspuren für Links und Rechts untergebracht werden, die nach dem Differenzverfahren auch noch einen Mitten- und Effektkanal beinhalteten (Dolby Stereo) (**Bild 14C**).
Im Rahmen der fortschreitenden Digitalisierung kam Anfang der Neunziger Jahre Dolby Stereo SR (Spectral Recording) auf den Markt, was nun Surround-Wiedergabe mit Subwoofer wie 5.1 oder 7.1 ermöglichte. Aus Platzgründen wird diese Toninformation zwischen den Perforationslöchern aufgebracht (**Bild 14B**). Da die Digitaltechnik empfindlich auf Aussetzer und Verschmutzungen der Tonspur reagiert, brachte Sony 1994 ein eigenes System heraus (SDDS), mit acht Kanälen und redundanten Tonspuren auf der linken und rechten Außenkante (**Bild 14A**).
Zu erwähnen ist auch noch das DTS-System (Digital Theater Sound). Hier befindet sich der Ton auf bis zu drei separaten CD-ROMs, die über eine Timecodespur mit dem Film verkoppelt sind (**Bild 14D**). Das funktioniert auch bei Filmriss oder mit fehlenden Bildern. Auch DTS ist ein mehrkanaliges Surroundsystem.
Inzwischen hat die Digitaltechnik den mechanischen Projektor abgelöst, optischen Film gibt es fast nur noch in speziellen Programm-Kinos. Letzter Schrei der Filmtontechnik ist Dolby Atmos, wo in Verbindung mit Deckenlautsprechern ein neues Rund-Um-Hörgefühl möglich wird. Bleibt abzuwarten, was noch alles auf uns zu kommt… ⏮

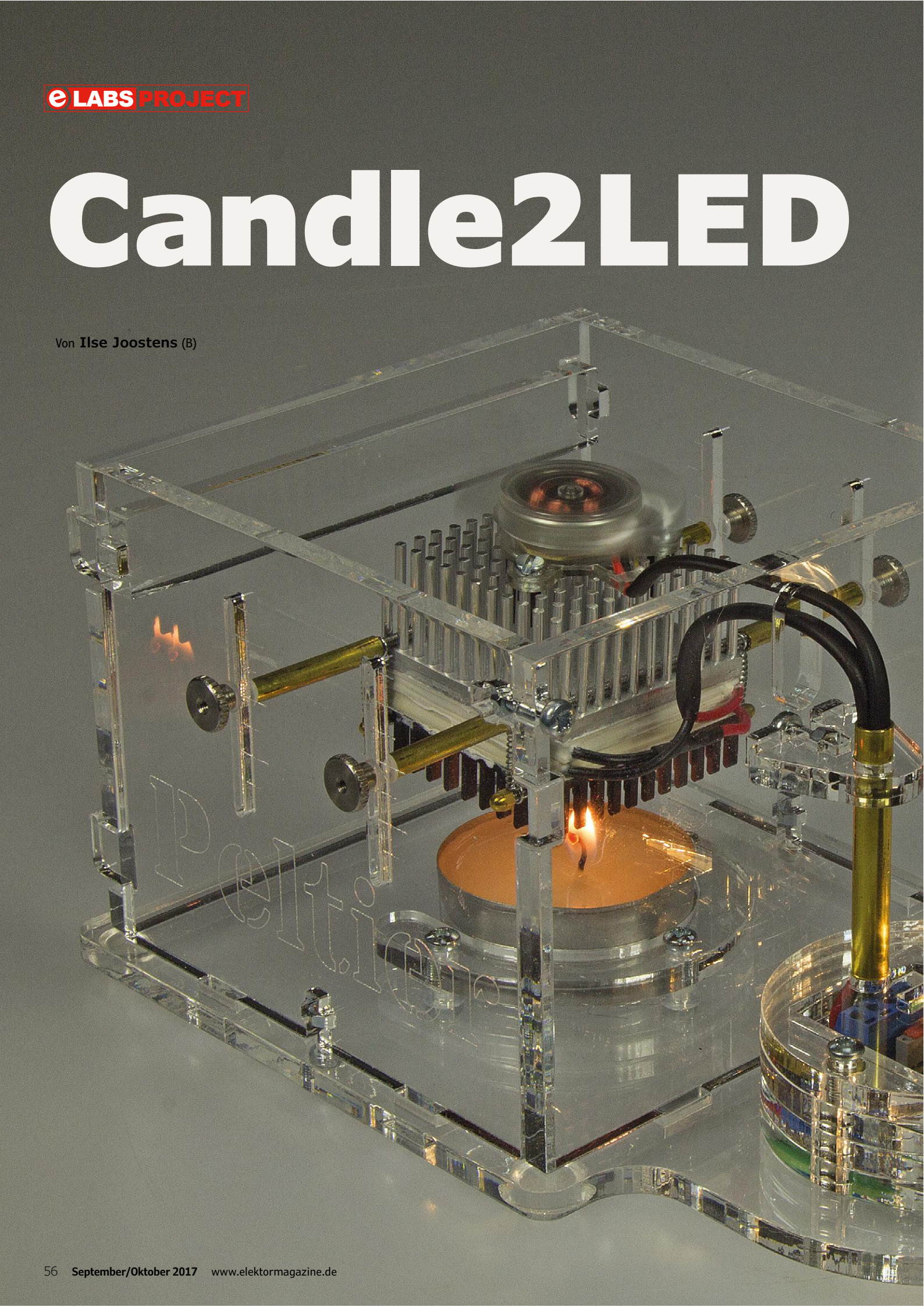
(160480)

ESTD 2004
www.elektor.tv
Retronik ist eine Rubrik, die antiker Elektronik und legendären Elektor-Schaltungen ihre Reverenz erweist. Beiträge, Vorschläge und Anfragen telegrafieren Sie bitte an Jan Buiting (editor@elektor.com).

Bildquellennachweis: Edison Historical Site, Wikipedia (unbekannter Urheber), Google pictures (unbekannter Urheber) und Fotos/Grafiken vom Autor. Im Bild 3 ist eine Person abgebildet von der keine Persönlichkeitsrechte zu ermitteln waren. Das Bild stammt aus den Jahren 1920.

# Candle2LED

Von **Ilse Joostens** (B)



# Kerzenlicht-Effizienzbooster

**Das Internet of Things gibt neue Anstöße, alternative Energiereservate zu erschließen. Gesucht sind Quellen, die gebundene Energie durch physikalische Prozesse in elektrischen Strom wandeln. Der „Candle 2 Light“-Booster setzt die Wärme einer Kerzenflamme in elektrische Energie um, so dass sogar das Lesen des Lieblingsromans bei elektrischem Kerzenlicht möglich ist!**

## INFOS ZUM PROJEKT

Peltier-Element
Alternative Energie
Step-up-Konverter

Einsteiger
➡ Fortgeschrittene
Experte

Ungefähr 2 Stunden

Bauteile-Paket,
Lötkolben,
lange Spitzzange

Etwa 100 €

Bevor die elektrische Beleuchtung zu Beginn des zwanzigsten Jahrhunderts zum Allgemeingut wurde, waren Kerzen und Petroleumlampen die gängigen Lichtspender. Die vermutlich ältesten Kerzen stammen aus China zur Zeit der Han-Dynastie, sie wurden aus dem Tran von Walfischen hergestellt. Seit ungefähr 1850 waren hauptsächlich tierische Fette die Grundsubstanzen, für hochwertige Produkte diente auch Bienenwachs als Ausgangsmaterial. Die industrielle Erdölgewinnung führte dazu, dass Paraffin zur Basis der Kerzenproduktion wurde. Anders als bei tierischen Fetten ist das Verbrennen von Paraffinen nahezu ohne Rauch möglich.

### Wenig Effizienz

Die Flamme einer durchschnittlichen Kerze aus heutiger Produktion hat eine Leistung von ungefähr 80 W. Bei einer Lichtausbeute von nur 0,16 Lumen etwa 32 W.

Wir haben darüber nachgedacht, ob sich die schlechte Energiebilanz ändern lässt. Mit der modernen Energie-Elektronik ist Verblüffendes machbar, warum nicht auch beim Wirkungsgrad von Kerzen? Teelichte sind vor allem in hohen Stückzahlen erstaunlich preiswert. Schon dies war ein Grund, die Energie für eine Beleuchtung eben nicht den Batterien, Akkus oder dem Stromnetz zu entnehmen. Die Teelichte können als alternative Energiequelle dienen!

### Realisierte Theorie

Unser thermo-elektrischer Generator arbeitet mit zwei in Reihe geschalteten Exemplaren des 40 · 40 mm großen Peltier-Elements TEC1-12706 (**Bild 1**). Die Peltier-Elemente sind mit Federn zwischen zwei Kühlkörpern eingeklemmt. Die Konstruktion ist höhenverstellbar über einem Teelichthalter angeordnet.

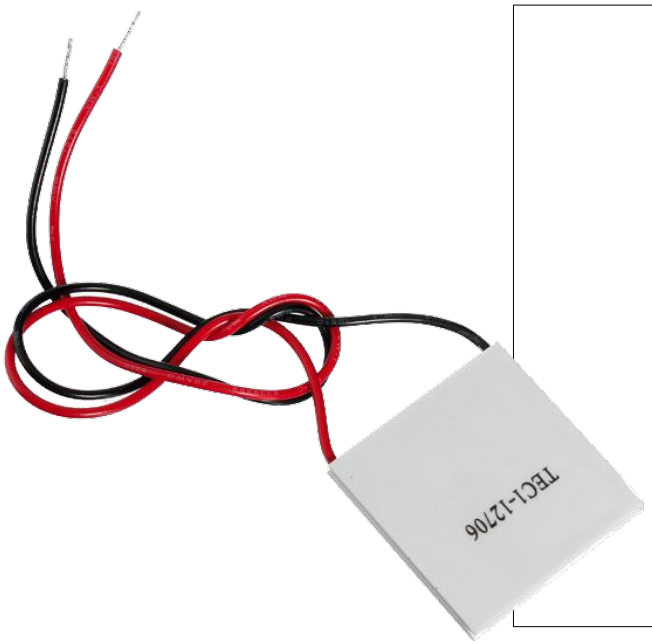



Bild 1. Das Peltier-Element TEC1-12706.

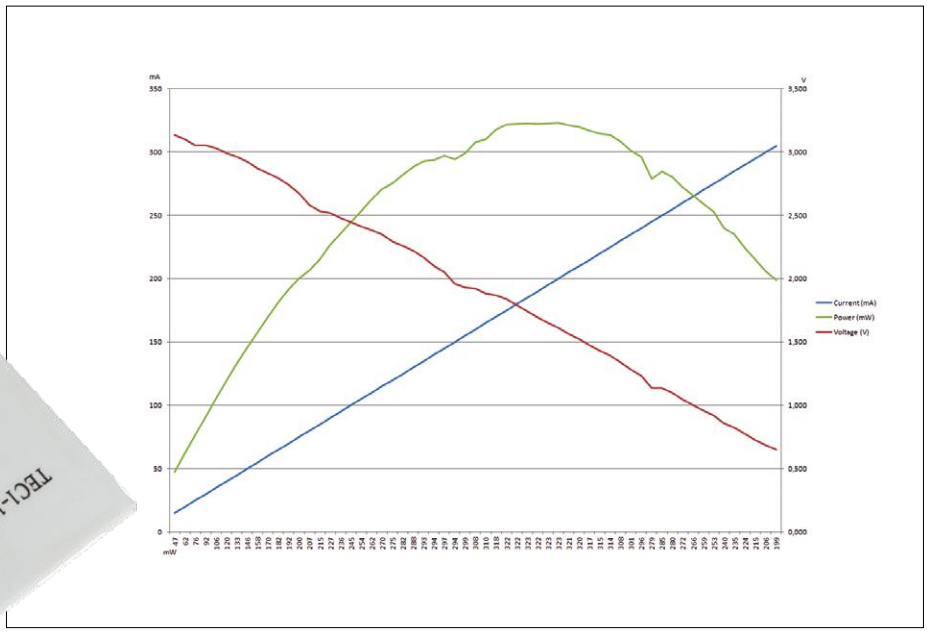



Bild 2. Verlauf von Spannung, Strom und Leistung. Der Maximum Power Point (MPPT) ist deutlich erkennbar.

## Eigenschaften

- Erzeugt elektrisches Licht aus Kerzenlicht
- Betriebsdauer 4 bis 5 Stunden pro Teelicht
- Attraktives transparentes Gehäuse (zwei Ausführungen erhältlich)
- Bauteile-Paket einschließlich LED-Lampe und Teelichten lieferbar

pro Watt wird die meiste Energie in Wärme umgesetzt. Der Wirkungsgrad ist verglichen mit einer konventionellen Glühlampe, die bekanntlich kein Vorbild an Effizienz ist, noch einmal um den Faktor 100 schlechter.

Die bekannten Teelichte, die auch bei diesem Projekt zum Einsatz kommen, brennen mit einem kürzeren Docht als lange Kerzen. Die Leistung von Teelichten liegt deshalb niedriger, abhängig von der Sorte des Paraffins beträgt sie nur

Der untere Kühlkörper mit den Abmessungen 40 · 40 · 40 mm verteilt die Wärme des Teelichts über die Oberflächen der Peltier-Elemente. Der obere Kühlkörper hat die doppelte Höhe, sein thermischer Widerstand liegt deshalb deutlich niedriger. Zusammen mit einem darauf montierten Lüfter sorgt er auf der kalten Seite der Peltier-Elemente für eine möglichst gute Kühlung.

Letztes Glied der Kette ist ein DC/DC-Spannungswandler, er setzt die

Spannung der Peltier-Elemente in eine für LED-Lampen passende Spannung um. Der Lüfter wird ebenfalls an der von den Peltier-Elementen erzeugten Spannung betrieben. Das mag auf den ersten Blick widersinnig erscheinen, doch die Praxis hat das Gegenteil bewiesen. Die für die LED-Lampe verfügbare Energiemenge ist mit dem Lüfter mindestens gleich oder höher als ohne Lüfter, gleichzeitig wird durch den Lüfter die Überhitzungsgefahr der Peltier-Elemente gemindert. Die internen Verbindungen des Peltier-Elements TEC1-12706 bestehen aus einer Wismut-Legierung, der Schmelzpunkt liegt bei 138 °C. Ein deutlicher Abstand von dieser Temperatur nach unten ist unbedingt notwendig.

Bei Messungen im Verlauf der Entwicklung dieses Projekts haben wir die Peltier-Elemente mit einem Drahtpotentiometer belastet. Die gemessene Ausgangsspannung abhängig von der Last ist in **Bild 2** grafisch dargestellt. Der Lüfter wurde an einer externen Spannungsquelle 3,3 V betrieben, so dass er die Messungen nicht beeinflussen konnte. Die Kerzenflamme befand sich bei maximaler Intensität etwa 5...10 mm unter dem unteren Kühlkörper, dabei stellte sich ein thermisches Gleichgewicht ein. Die Spannung an den Peltier-Elementen lag ohne Belastung bei 3,2 V. Da sich Peltier-Elemente (und auch andere alternative Energiequellen) wie Spannungsquellen mit hohen Innenwiderständen verhalten, brach die Spannung bei steigender Belastung ein. In der Grafik ist auch die gewonnene Leistung (Spannung · Strom) dargestellt. Dieser Kurve lässt sich entnehmen, dass die maximale Leistung etwa 320 mW im Spannungsbereich 1,6...1,7 V beträgt. Dort befindet sich der so genannte *Maximum Power Point* (MPPT). Wenn die Kerzenflamme weniger intensiv leuchtet, beträgt die offene Klemmenspannung nur noch etwa 2,2 V, die Leistung im MPPT geht auf ungefähr 180 mW zurück. Ohne Lüfter sind nur Leistungen von 80...100 mW erreichbar. Theoretisch beträgt der Wirkungsgrad eines thermoelektrischen Generators 5...8 %, so dass ein Teelicht mit der Wärmeleistung 32 W eine elektrische Leistung von 1,6...2,56 W liefern kann. In der Praxis fällt die Ausbeute an elektrischer Energie niedriger aus, weil Wärmeleistung an die Umgebung verloren geht und die verwendeten Peltier-Elemente nicht für den Einsatz in thermoelektrischen Generatoren konstruiert sind. Die

## Der Seebeck-Effekt

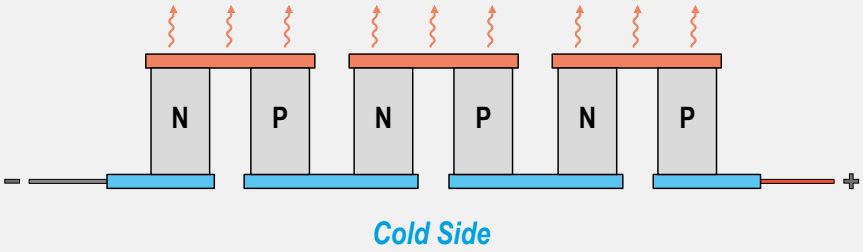
Jeder gestandene Elektroniker kennt vermutlich das Prinzip des Thermoelements. Weniger bekannt dürfte die Thermosäule sein, die aus mehreren in Reihe geschalteten Thermoelementen besteht und zum präzisen Messen geringer Temperaturdifferenzen eingesetzt wird. Bekannte Anwendungen sind kontaktlose Infrarot-Thermometer und elektronische Fieberthermometer. Nur nebenbei: Wärmebildkameras verwenden so genannte Mikrobolometer, ihr Funktionsprinzip ist ein völlig anderes.

Der baltisch-deutsche Physiker Thomas Johann Seebeck entdeckte im Jahr 1821, dass ein Ring aus zwei unterschiedlichen Metallen eine Kompassnadel bewegt, wenn zwischen den beiden Schweißstellen eine Temperaturdifferenz besteht. Ihm blieb jedoch verborgen, dass in dem Ring ein Strom fließt, deshalb nannte er seine Entdeckung „Thermomagnetischer Effekt". Erst der dänische Physiker Hans Christian Ørsted erkannte die tatsächlichen Zusammenhänge, von ihm stammt der Begriff „Thermo-Elektrizität".

Es vergingen 13 Jahre, bis der französische Uhrmacher Jean Athanase Peltier entdeckte, dass der Seebeck-Effekt umkehrbar ist. Wenn ein Strom durch einen aus zwei Metallen zusammengeschweißten Draht fließt, steigt die Temperatur auf einer Seite der Schweißstelle, während die Temperatur der anderen Seite sinkt. Erst ab ungefähr Mitte des letzten Jahrhunderts wurde der Peltier-Effekt dank der damals aufstrebenden Halbleiterfertigung technisch genutzt. Der Effekt tritt nämlich nicht nur bei unterschiedlichen Metallen, sondern auch bei Halbleitern in Erscheinung. Das konnte Jean Athanase Peltier zu seiner Zeit noch nicht ahnen.

Ein Peltier-Element ist eine Festkörper-Wärmepumpe, die mit elektrischer Energie Wärme von einer Seite zur anderen Seite entgegen einem Temperaturgradienten transportiert. Das Bauelement besteht aus zwei Platten aus thermisch leitendem, jedoch elektrisch isolierendem keramischen Material, meistens Aluminiumoxid, $Al_2O_3$. Das Halbleiter-Material ist zwischen den als Träger dienenden Platten eingebettet. Verwendet wird meistens n- und p-dotiertes Wismut(III)Tellurid, die einzelnen Bereiche sind über Platinenbahnen auf dem keramischen Träger miteinander verbunden. Auf der kalten Seite fließt der Strom vom p- zum n-dotierten Material, auf der warmen Seite entgegengesetzt (konventionelle Stromrichtung).

Wegen des schlechten Wirkungsgrads von Peltier-Elementen werden sie hauptsächlich zur Kühlung in Anwendungen eingesetzt, die auf engem Raum mit wenig Kühlleistung auskommen. Beispiele sind die bekannten Kühlboxen, und auch die Bildsensoren einiger Digitalkameras werden mit Peltier-Elementen gekühlt.




Querschnitt durch ein Peltier-Element.

Der Effekt lässt sich umkehren: Wenn eine Seite eines Peltier-Elements erwärmt und die andere Seite gekühlt wird, liefert das Peltier-Element elektrische Energie. Entstanden ist ein so genannter thermoelektrischer Generator, kurz TEG.

Das Einsatzfeld thermoelektrischer Generatoren sind vorwiegend die Satelliten, früher dienten sie auch zur Stromversorgung unbemannter Anlagen in abgelegenen Regionen. Bei solchen Anwendungen ist die Wärmequelle ein radioaktives Isotop, meistens Plutonium 238 oder Strontium 90, die Wärme entsteht durch radioaktiven Zerfall. Zurzeit beschäftigt sich die Forschung unter anderem mit dem *Automotive Thermoelectric Generator* (ATEG), bei dem ein Teil der von Verbrennungsmotoren abgegebenen Wärme in elektrische Energie für die Verbraucher an den Bordnetzen umgewandelt wird.

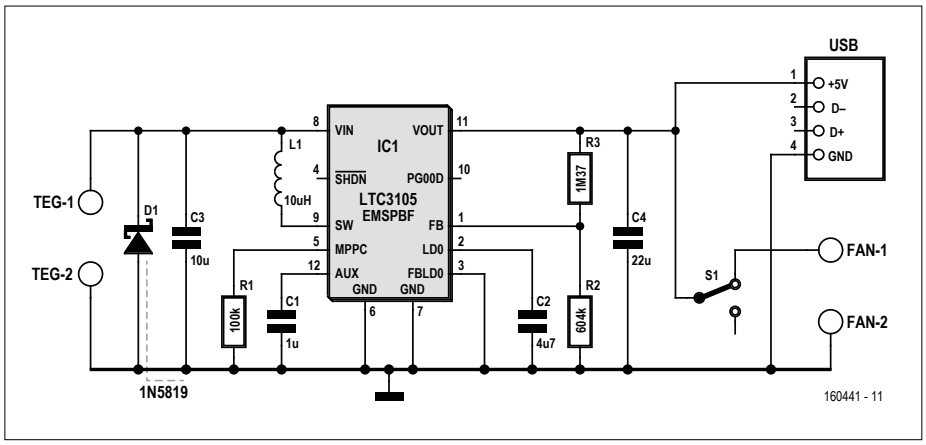



Bild 3. Der LTC3105 ist ein Step-up-DC/DC-Wandler mit einstellbarem MPPT.

## STÜCKLISTE

**Widerstände (1 %, 0,125 W, SMD 0805):**
R1 = 100 kΩ
R2 = 604 kΩ
R3 = 1,37 MΩ

**Kondensatoren (10 V, X7R MLCC):**
C1 = 1 µF, 10 V, 0805
C2 = 4,7 µF, 10 V, 0805
C3 = 10 µF, 10 V, 0805
C4 = 22 µF, 10 V, 1206

**Halbleiter:**
D1 = 1N5819HW-7-F SOD-123
IC1 = LTC3105EMS#PBF
TEC1, TEC2 = TEC1-12706, 40 · 40 mm

**Außerdem:**
L1 = 10 µH 850 mA, abgeschirmt WE-MAPI (Würth 74438335100)
K1, K2 = Kabelschraubklemme, 2-polig, RM 3,5 mm
K3 = USB-Buchse, Typ A stehend (Würth 614004185023)
S1 = Schiebeschalter, SPDT, ON-ON (Würth 450301014042)
LED-Leuchte „Jansjö USB“ (IKEA 702.912.32)

**Mechanische Bauteile:**
1 · Kühlkörper 40 · 40 · 20 mm, Aluminium, Fischer ICK S Series (Farnell 1850058)
1 · Kühlkörper 40 · 40 · 10 mm, Aluminium schwarz eloxiert, Fischer ICK BGA Series (Farnell 1850047)
2 · Gewindestange M2, L = 118 mm
2 · Gewindestange M2, L = 51 mm
4 · Hutmutter M2 (microschroeven.nl, MD-DopMoer -M2 - ME)
4 · Rändelmutter M2, hoch, RVS (Fabory 51830020001)
4 · Zugfeder 0,5 · 3,5 · 12 mm (Fabory 17902053012)
4 · Messingrohr, D = 4 mm · 0,5 mm, L = 26 mm
8 · Unterlegscheibe M2, Polyamid

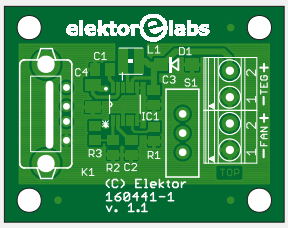



Bild 4. Doppelseitige Platine für den Effizienzbooster.

Bild 5. Ansicht der aufgebauten Platine von oben.

1 · Lüfter, transparentes Gehäuse, offen, D = 35 mm, für VGA-Karte (4894462470268)
3 · Schraube M3, L = 6 mm, Zylinderkopf/Schlitz (M36 CSSTMCZ100)
1 · Schrumpfschlauch 3 mm, Schrumpfung 1:2
1 · Schrumpfschlauch 1,6 mm, Schrumpfung 1:2
1 · Wärmeleitpaste auf Silikon-Basis (Tube)
5 · Gummifuß, selbstklebend (TME RI-RBS-12)
14 · Schraube M3, L = 12 mm, Stahl verzinkt, Pozidriv, DIN 7985A
3 · Schraube M3, L = 16 mm, Stahl verzinkt, Pozidriv, DIN 7985A
13 · Mutter M3, Sechskant, Stahl, DIN 934
7 · Abstandshalter 3 mm, Polyamid
4 · Unterlegscheibe M3, Polyamid, DIN 125A
3 · Wellscheibe M3
1 · Messingrohr, D = 6 mm · 0,5 mm, L = 45 mm
2 · Abstandshalter F/F, M3, L = 15 mm
1 · Gehäuse, Acryl transparent, extrudiert 5 mm, lasergeschnitten

Typen mit dem Präfix „TEG1…“ vertragen wesentlich höhere Temperaturen, sie sind für solche Zwecke vorgesehen, haben aber erheblich höhere Preise.

### Energie-Elektronik

Beim Projektstart hatten wir die Absicht, die Elektronik vollständig mit bedrahteten Bauelementen zu realisieren, um den Aufbau möglichst nachbaufreundlich zu gestalten. Leider werden viele integrierte Bausteine nur noch in SMD-Bauform produziert, und bedrahtete Elkos anstelle platzsparender SMD-MLCC-Ausführungen hätten deutlich größere Abmessungen der Platine erfordert. Deshalb haben wir unsere Absicht zugunsten der SMDs aufgegeben.

Wichtigstes Bauteil der Schaltung in **Bild 3** ist der LTC3105, ein Step-up-DC/DC-Wandler, bei dem sich der MPPT einstellen lässt. Diese Eigenschaft prädestiniert das IC für den hier vorgesehenen Zweck. Der Eingangsspannungsbereich beginnt bei 250 mV, so dass der LTC3105 auch bei vielen *Energy-Harvesting*-Anwendungen zum Einsatz kommt.

Die Schaltung lehnt sich weitgehend an das Beispiel aus dem Datenblatt des LT3105 an. Mit R1 = 100 kΩ ist der MPPT auf ungefähr 1 V eingestellt. Dieser Wert liegt absichtlich niedriger als der bei intensivem Kerzenlicht erreichte tatsächliche MPPT. Die dadurch bedingten Energieverluste belaufen sich zwar auf etwa 50 mW bei maximaler Flammenintensität (270 mW statt 320 mW), doch der Wandler startet stabiler, unmittelbar nach Anzünden des Teelichts.

Es ist zwar schaltungstechnisch möglich, eine Art MPPT-Tracking hinzuzufügen, indem die Kühlkörper-Temperatur gemessen wird und eine Rückführung zum LTC3105 stattfindet, doch wegen der hohen Komplexität und des eher geringen Nutzens haben wir darauf verzichtet. Hinzu kommt, dass während der vier bis fünf Stunden, die ein gängiges Teelicht brennt, die maximale Intensität nur über eine begrenzte Zeit erreicht wird.

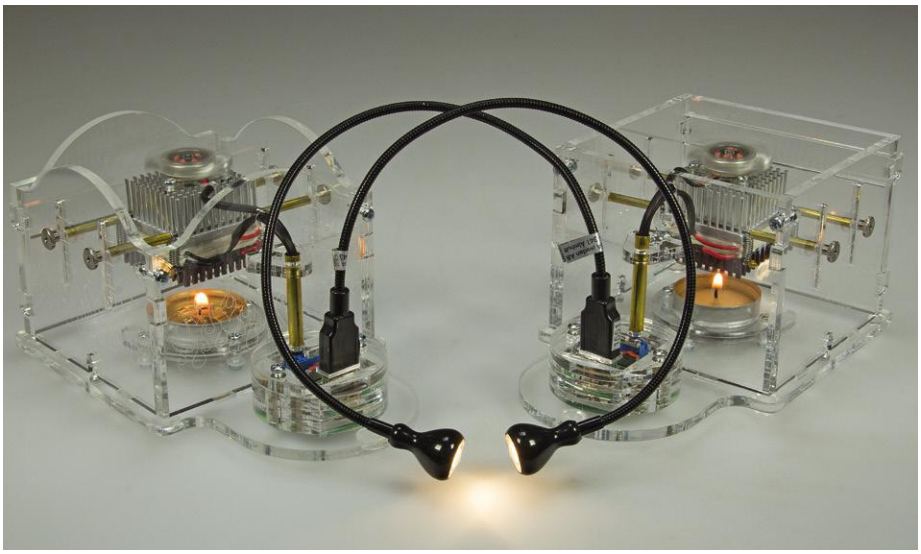
Die MLCC-Kondensatoren haben verglichen mit konventionellen Elkos extrem kompakte Bauformen, in einem SMD des Typs 0805 sind Kapazitäten bis 10 µF untergebracht. Ein Minuspunkt des verwendeten Dielektrikums X7R ist das Sinken der Kapazität beim Steigen der anliegenden Spannung. Vergleichbar ist dieser Effekt mit einer Blumenvase, die sich nach oben verengt. Wenn der Wasserstand steigt, steigt auch der Druck

auf den Vasenboden, es kann jedoch weniger Wasser hinzu gegossen werden. Der Effekt ist nicht vernachlässigbar, die Kapazität sinkt bei 3,3 V bereits um rund 20 %. Aus diesem Grund wurde die Kapazität des Ausgangskondensators C4 von zuerst 10 µF auf 22 µF erhöht. Der Wert 22 µF ist nur in der Bauform 1206 erhältlich.

Die Stromaufnahme der LED-Lampe, die hier zum Einsatz kommt, beträgt 55 mA bei der Spannung 5 V, was der Leistung 0,275 W entspricht. Damit liegt sie deutlich über der Leistung 0,15 W, die im Produktdatenblatt angegeben ist. Unser thermoelektrischer Generator kann diese Leistung nicht liefern. Bei 3,3 V sinkt die Stromaufnahme der Lampe auf 39 mA, die Leistung beträgt nur noch 130 mW. Die Lichtintensität wird um 30 % reduziert, sie reicht jedoch noch aus, um ein Buch zu lesen. Aus den vorstehenden Gründen haben wir die Ausgangsspannung des DC/DC-Wandlers auf 3,3 V begrenzt. Bei Spannungen im Bereich 3,0...3,3 V entfällt auf den Lüfter eine Leistung von etwa 95...120 mW. Unser thermoelektrischer Generator ist folglich den Anforderungen gewachsen, auch wenn bei schwacher Kerzenflamme die Ausgangsspannung auf etwa 3,0 V sinkt. Der Lüfter lässt sich mit einem Schiebeschalter abschalten. Das kann von Vorteil sein, wenn bei schwacher Kerzenflamme kurzzeitig eine höhere Lichtintensität erwünscht ist. Die zuvor vom Lüfter aufgenommene Energie ist dann für die LED-Lampe verfügbar. Dieser Effekt ist allerdings von nur kurzer Dauer, denn ohne Lüfter steigt die Temperatur des oberen Kühlkörpers an. Die Temperaturdifferenz zwischen den Seiten der Peltier-Elemente sinkt, so dass die entnehmbare Leistung ebenfalls sinkt. Wenn der Lüfter längere Zeit abgeschaltet bleibt, muss unbedingt dafür gesorgt werden, dass sich die Peltier-Elemente nicht überhitzen. Das kann durch Erhöhen des Abstands zwischen der Kerzenflamme und dem unteren Kühlkörper geschehen.

Unser thermoelektrischer Generator ist zum Laden von Mobiltelefon-Akkus, auch nach Erhöhen der Spannung auf 5 V, ausdrücklich *nicht* geeignet, weil die Leistung nicht ausreicht. Das Laden würde mehrere Tage dauern, natürlich vorausgesetzt, dass das Mobiltelefon vollständig abgeschaltet ist. Auf YouTube wird in mehreren Clips demonstriert, dass die Ladeanzeige eines Mobiltelefons reagiert. Doch das bedeutet noch nicht, dass der Akku tatsächlich geladen wird.

## Romantisches Kerzenlicht und helle LED-Beleuchtung reichen sich die Hand



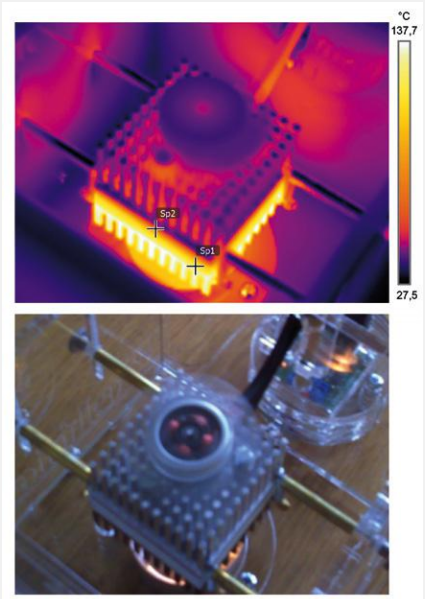
### Aufbau und Betrieb

Der erste Schritt ist der Aufbau der Platine, **Bild 4** und **Bild 5** zeigen die Details. Montieren Sie zuerst die SMD-Bauelemente, danach folgen die USB-Buchse, der Schiebeschalter und die Kabelschraubklemmen. Um die aufgebaute Platine zu testen, können Sie die Anschlüsse TEG+ und TEG– über einen Widerstand 10 Ω mit einer Spannungsquelle 2 V verbinden (eventuell genügt eine 1,5-V-Batterie AA). An den Anschlüssen des Ventilators müssen dann bei geschlossenem Schiebeschalter 3,3 V liegen. Zwischen TEG+ und TEG– muss abhängig von der Belastung der Ausgangsspannung mindestens 1 V messbar sein. Dies ist ein Zeichen dafür, dass die MPPT-Funktion des DC/DC-Wandlers arbeitet.

Die nächste Aktion ist die Montage des thermoelektrischen Generators. Schrauben Sie die langen Gewindestangen mit

## Thermisches Verhalten

Wir haben den Candle 2 LED Booster mit einer Wärmebildkamera FLIR Modell E40 während des Betriebs aufgenommen, das Ergebnis zeigen die Fotos (oben: Infrarot, unten: Normallicht). Das Infrarot-Foto lässt das Temperaturgefälle vom unteren zum oberen Kühlkörper, zwischen denen die Peltier-Elemente eingeklemmt sind, deutlich erkennen. Der laufende Lüfter trägt nicht unwesentlich dazu bei. Weniger deutlich ist sichtbar, dass auch die Außenseite des Gehäuses auf Temperatur ist, was eine (vorsichtige!) Fingerprobe bestätigen kann.



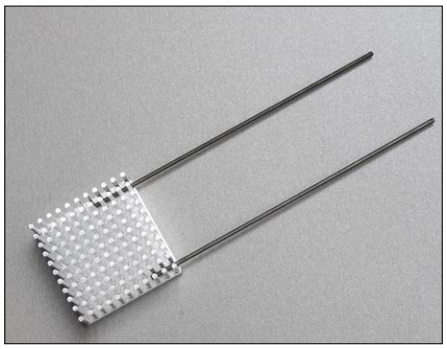


Bild 6. Die Enden der langen Gewindestangen werden zwischen die Kühlkörper-Pfosten gedreht.

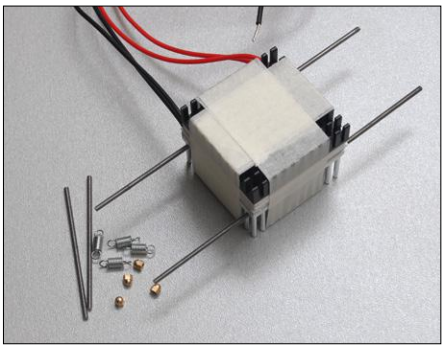


Bild 7. Während der Montage wird die Konstruktion mit Abdeckklebeband fixiert.

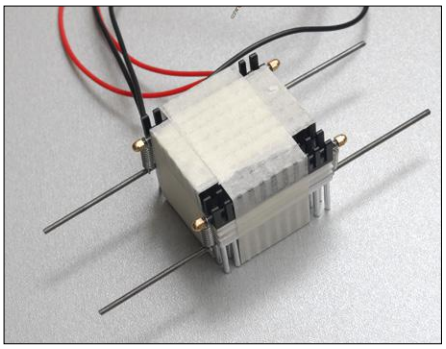


Bild 8. Die Zugfedern pressen die Kühlkörper gegen die Peltier-Elemente.

drehenden Bewegungen, wie in **Bild 6** dargestellt, zwischen die Pfosten des oberen Kühlkörpers. Nicht von oben hineindrücken! Die Peltier-Elemente legen Sie auf den oberen, sparsam mit Wärmeleitpaste bestrichenen Kühlkörper, dabei müssen Sie auf die Orientierung der Peltier-Elemente achten. Auf den Stapel folgt dann der untere Kühlkörper. Fixieren Sie den Stapel vorübergehend, wie in **Bild 7** gezeigt, über Kreuz mit Abdeckklebeband. Bringen Sie die kurzen Gewindestangen an und spannen Sie mit einer langen Spitzzange die vier Zugfedern darüber (**Bild 8**). Montieren Sie schließlich den Lüfter.

Nun steht die Montage des transparenten PMMA-Gehäuses an, es ist in zwei Ausführungen lieferbar. Das Gehäuse dient gleichzeitig als Windschutz für die Kerzenflamme. Montieren Sie den thermoelektrischen Generator im Gehäuse, anschließend montieren Sie die Spannungswandler-Platine. Zum Schluss stecken Sie die LED-Lampe in die USB-Buchse. **Bild 9** zeigt, wie der fertige Aufbau aussieht.

Stellen Sie jetzt die Höhe des thermoelektrischen Generators mit den seitlichen Rändelmuttern so ein, dass der Abstand zur Flammenspitze etwa 15 mm beträgt. Schalten Sie mit dem Schiebeschalter den Lüfter ein, und setzen Sie das brennende Teelicht vorsichtig in den Halter. Nach einigen Minuten leuchtet die LED-Lampe schwach auf. Warten Sie noch eine bis zwei Minuten, während die Helligkeit langsam steigt, und stoßen Sie dann das Lüfterrad entgegen dem Uhrzeigersinn an. Das Lüfterrad sollte nun rotieren, anderenfalls stoßen Sie es nach kurzer Wartezeit noch einmal an. Falls sich das Lüfterrad nicht dreht, prüfen Sie die Stellung des Schiebeschalters. Das Lüfterrad muss angestoßen werden, weil der Lüfter für die Spannung 12 V ausgelegt ist, er wird hier außerhalb seines normalen Spannungsbereichs betrieben. Bei seiner Nenndrehzahl wäre der Energiebedarf zu hoch, im Luftstrom würde die Teelichtflamme stark flackern und es würden unerwünschte Geräusche entstehen.

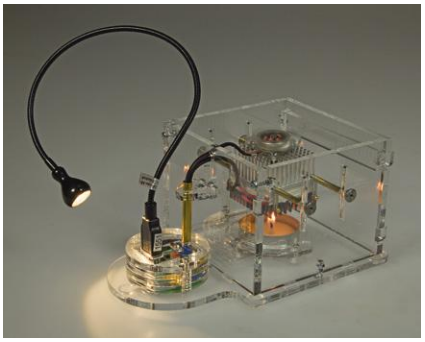
Wenn der Lüfter zu drehen beginnt, flackert die LED-Lampe, weil der thermoelektrische Generator noch nicht die maximale Energie liefert und elektrische Störsignale die Elektronik beeinträchtigen. Das Flackern hält nur wenige Minuten an, abhängig von der Intensität der Teelichtflamme. Ist das Paraffin noch nicht vollständig geschmolzen, kann diese Zeit kürzer sein. Während des Betriebs kann die Höhe des thermoelektrischen Generators abhängig von der Teelichtflamme von Zeit zu Zeit nachgestellt werden. Die Höhe ist wenig kritisch, ein Abstand von 10...20 mm zur Flammenspitze ist meistens optimal.

Zum Schluss noch ein sehr wichtiger Hinweis: Da der Candle 2 LED Booster mit einer offen Flamme betrieben wird, darf er wegen potentieller Brandgefahr niemals unbeaufsichtigt gelassen werden! ◀

(160441)gd

**EINKAUFSZETTEL**

→ 160144-71/72
Bauteile-Paket



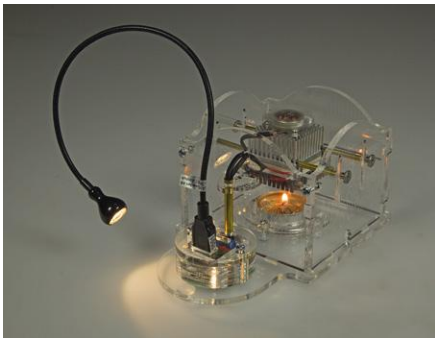
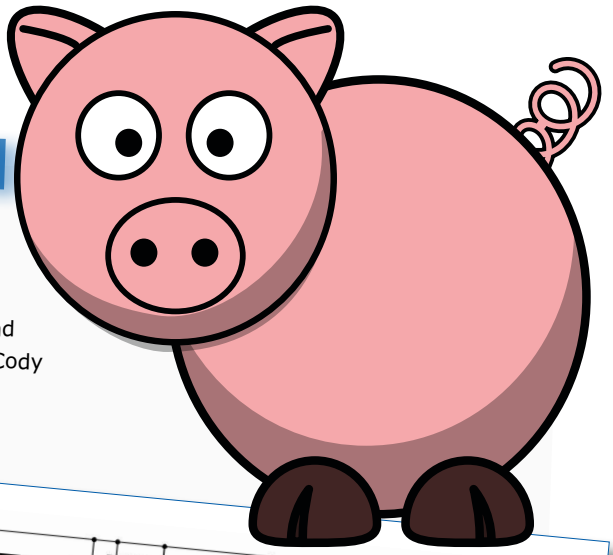


Bild 9. Das Gehäuse ist in zwei Ausführungen erhältlich.

# HomeLab Helicopter

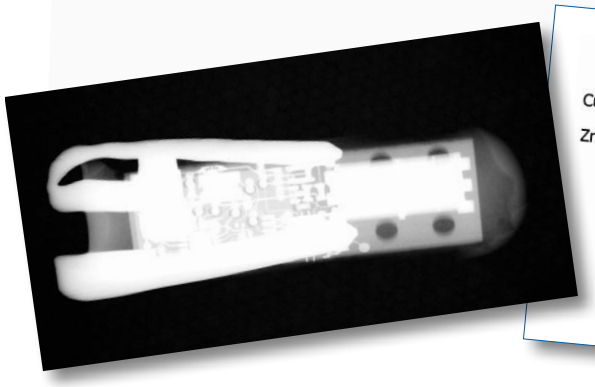
Zusammengestellt von **Clemens Valens** (Elektor-Labor)

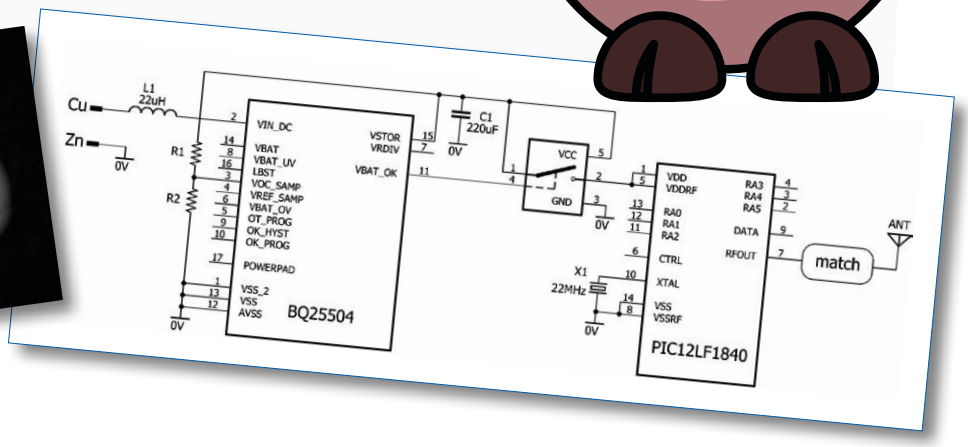


## Einnehmbare Elektronik

Diese 40×10 mm große elektronische Pille misst die Temperatur im Inneren eines Schweinekörpers und sendet für sechs Tage fünf Mal pro Minute die erfassten Daten zusammen mit einigen Leistungsindikatoren kabellos an einen Empfänger. Die Energie erhält der Sensor mit Hilfe von Kupfer- und Zinkelektroden aus der Magenflüssigkeit (Röntgenbild von Jenny Haupt, Cody Cleveland und Phillip Nadeau).

www.nature.com/articles/s41551-016-0022





**MediaMarkt eröffnet Drive-thru-Laden für Elektronikgeräte im niederländischen Tilburg.**

# Ein Muss in jedem Heimlabor

Virile und Virale: Diese äußerst männliche Uhr funktioniert auch als Zigarettenanzünder. Es ist alles elektrisch und funktioniert ohne Gas, was bedeutet, dass sie auch windsicher und sicher ist. Sie ist großartig zum Entzünden Ihres Butan-Gas-Lötbrenners, aber seien Sie vorsichtig, dass Sie nicht Ihr haariges Handgelenk (oder die Uhr) verbrennen. Aufgeladen wird sie über USB. Der Clou sind drei kleine, dekorative, aber nicht funktionierende (!) Sub-Zifferblätter. Es ist einer jener Gegenstände, die du als cooles Geschenk für einen brüderlichen Freund kaufst, aber am Ende selbst behältst. Verfügbar im ganzen Netz - suche nach der „USB Lighter Watch".

**Google bestellt für 1.000.000.000.000 Won (790.000.000 €) OLED-Displays bei LG.**

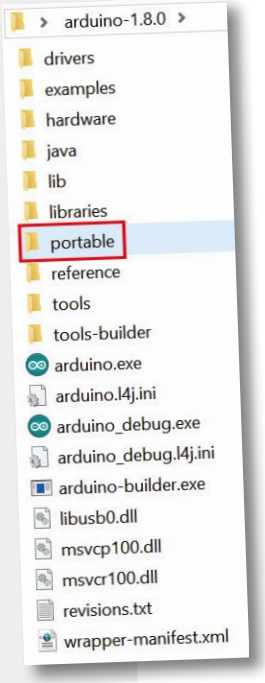
## Portable Arduino

Wie die meisten (Windows-) Anwendungen legt auch die Arduino-IDE ihre Zwischendateien und Arbeitsmappen an festen Orten auf der Computerfestplatte an. Für die meisten Benutzer, die immer nur auf ihrem eigenen Computer arbeiten, spielt es keine Rolle, aber wenn man einen öffentlichen oder Uni-Computer benutzen muss, auf dem man keine Board-Packages oder Bibliotheken installieren darf, kann dies ein Problem darstellen. Die Lösung ist, die Arduino-IDE portabel zu machen. Eine tragbare Arduino-IDE enthält alles, was man wünscht, in einem Ordner.

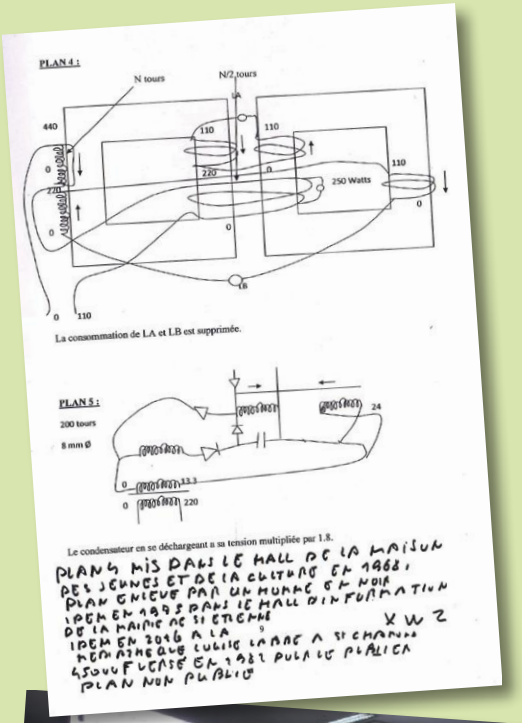
Die IDE portabel zu machen ist überraschend einfach. Alles, was Sie tun müssen, ist im Installationsordner (mit der arduino.exe) einen Ordner namens „portable" anzulegen. Beim nächsten Start der IDE wird er automatisch erkannt und als Speicherort verwendet. Alle Board-Pakete und Bibliotheken, die Sie installieren, und auch alle Zwischendateien werden dort untergebracht. Alles befindet sich dann ordentlich in einem einzigen Ordner, den Sie auf einen kleinen Flash-Speicher kopieren und verwenden können, wo und wann es Ihnen beliebt. Die portable IDE ist auch ein guter Weg, um verschiedene Arduino-Installationen auf einem Rechner getrennt zu halten. Bibliotheken und Board-Pakete der Arduino-IDE werden ständig aktualisiert. Wenn man stets die aktuelle IDE installiert, wird zwar jedes neue Projekt davon profitieren, aber es passiert auch leicht, dass ein altes Projekt nicht mehr funktioniert. Mit einer portablen IDE werden Projekte aber leicht archiviert.

Der portable Ordner ist selbst tragbar und kann in andere IDE-Installationen kopiert werden. Dies ermöglicht eine schnelle Einrichtung von identischen IDEs auf mehreren Computern, was Lehrer und Administratoren sicherlich zu schätzen wissen.

https://www.arduino.cc/en/Guide/PortableIDE



# Freie Energie



Wir haben mit der Post ein interessantes Angebot erhalten, eine Sammlung von 15 „Freien Energie"-Projekten. Falls Sie unvertraut mit der Materie sind, bei der Freien Energie geht es um Perpetuum Mobiles, kalte Fusionsgeneratoren, torusbasierte Generatoren, Rekonstruktion extraterrestrischer Technologie und andere allgemein unbewiesene, kostengünstige Energiequellen. Der avisierte Preis von 200 € (inklusive 65 € für die obligatorische Mitgliedschaft in der *Fondation Pour La Libre Energie)* ist sicherlich nicht zu viel für eine großartige Sammlung revolutionärer, minderwertiger, unverständlicher Zeichnungen und unleserlicher Großbuchstabentexte. Ich meine, wer möchte nicht eine monopolare Lichtmaschine bauen, die nicht unter dem Lenzschen Gesetz leidet? Alles, was man tun muss, ist die magnetische Ladung eines Elektronenbündels umkehren und schon kann es losgehen! (magnetische Ladung?)
Videos von spannenden Free-Energy-Projekten, die dazu bestimmt sind, skeptische Zeitgenossen zu überzeugen, sind auf YouTube reichlich vorhanden. Hier zwei Links zum Start:

https://youtu.be/Qrw6Xj5a0nM

https://youtu.be/wnJpMX-GXcg

## Du großes Gestirn! Was wäre dein Glück, wenn du nicht die hättest, welchen du leuchtest!

Vor ein paar Jahren hatte jede Website einen Besucherzähler. Heute sind sie meistens verschwunden, was aber nicht bedeutet, dass die Besucher nicht mehr gezählt werden. Die Zahl ist nur privat im Backend der Website versteckt. Ein Besucher-Zähler gibt einen Hinweis auf die Popularität der Website, hilft aber vor allem, das Ego der Webseiten-Besitzer zu steigern (vor allem, wenn die Besucherzahl schnell ansteigt). Allerdings geht, wenn die Besucherzahl nur im Statistikmodul der Website sichtbar ist, ein wenig ihre Ego-Verstärkungseigenschaft verloren. Vor ein paar Jahren entwickelte Bogdan Raducanu, auch bekannt als Electro Bob, ein großartiges Gadget, um diesen Mangel zu beheben: Dieses kleine Kästchen ermittelt ganz alleine die Zahl der Besucher und zeigt sie auf seinem 8-stelligen 7-Segment-Display. Stellen Sie die Ego Box irgendwo sichtbar auf und nehmen Sie einen Ego-Shoot, wann immer Sie ihn brauchen (ohne sich dabei in ihre Webseite zu loggen)!
Wahrscheinlich ist es nicht zu kompliziert, die Software zu modifizieren und die Zahl Ihrer Social-Media-Follower oder eine andere interessante Statistik anzuzeigen. Wenn Sie, wie ich, keine Anhänger oder Besucher haben, können Sie sich immer noch die Tageszeit anzeigen lassen; die Ego-Box verfügt ja über genügend Digits.
Auch für alle, die nicht an der Ego-Box interessiert sind, bietet die Webseite von Electro Bob viele gute Sachen, die es zu entdecken gilt...
www.electrobob.com/ego-box/

Die Ego Box zeigt die Anzahl der Besucher Ihrer Website an (Bogdan Raducanu, www.electrobob.com).

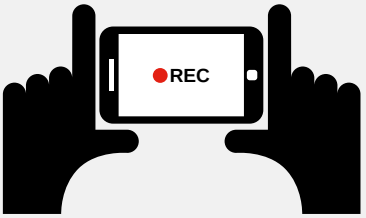
**Möchten Sie mitmachen? Bitte senden Sie Ihre Kommentare, Anwendungen, Tipps und Tricks an labs@elektor.com!**

# Machen Sie mit bei der Elektor-Videoolympiade

Haben Sie ein Elektor-Projekt aufgebaut oder einen Elektor-Artikel gelesen? Etwas optimiert, etwas modifiziert, etwas kaputtgemacht? Fanden Sie es gut? Oder furchtbar? Das ist Ihre Chance, uns zu informieren und dafür belohnt zu werden. Filmen Sie Ihre Elektor-Erfahrung, nehmen Sie an der Elektor-Videoolympiade teil und gewinnen Sie den Hauptpreis!

## Bleibe du selbst!

Versuchen Sie nicht, bekannte Video-Blogger nachzuahmen. Wir mögen einfache, einprägsame, amüsante Videos über "etwas mit Elektor" - mit viel Elektronik und technischem Zeugs. Humor darf gern sein, aber übertreiben Sie es nicht. Das Ziel ist es, so viele "Likes" wie möglich von den Zuschauern zu sammeln. Achten Sie auf hohe Bildqualität (vorzugsweise HD 1920×1080 Pixel). Verwenden Sie keine Musik, für die Sie keine Rechte haben. Achten Sie darauf, dass im Video klar und verständlich gesprochen wird.



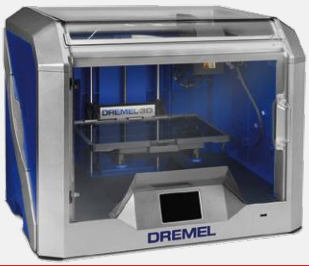

## Quick Start Guide

Drehen und bearbeiten Sie Ihren Film für den Wettbewerb. Senden Sie ihn vor dem 15. November 2017, 18 Uhr MEZ, über die Cloud WeTransfer an tv@elektor.com. Bitte senden Sie uns keine Beiträge per E-Mail! Wenn Ihr Video die Spielregeln einhält, wird es auf Elektor TV (www.elektor.tv) veröffentlicht, wo es angesehen und geliked werden kann. Die 15 Videos mit den meisten "Likes" werden von einer Jury beurteilt, um die Gewinner des Haupt- und der kleinen Preise zu bestimmen. Jeder Regisseur, dessen Film wir veröffentlichen, wird mit dem Buch "Analogue Video" von Angelo La Spina belohnt.

*„Dabei sein is alles?*
*Viel wichtiger ist das Gewinnen!"*

## Was gibt es zu gewinnen?

**Hauptpreis**
3D-Drucker
DREMEL 3D
Idea Builder 3D40
(im Wert von 1.449 €)



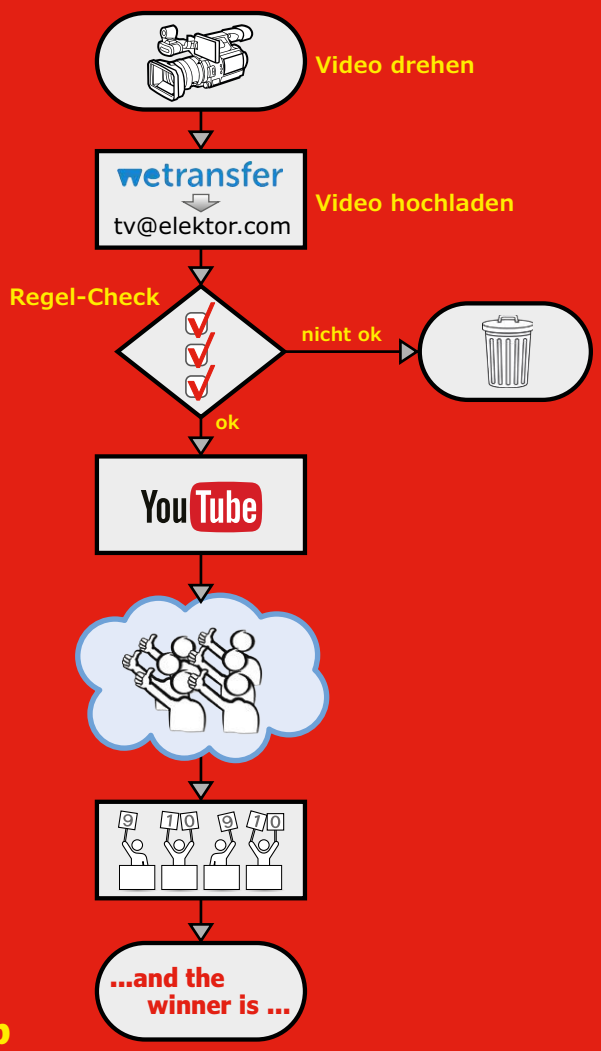

**Erster Preis**
Oszilloskop PicoScope 2208B MSO, 100 MHz
(im Wert von 1.099 €)

**Zweiter Preis**
LabNation SmartScope Maker Kit (im Wert von 300 €)

# Dabei sein ist gut – gewinnen ist besser!

## Spiel nach Regeln

1. Jeder kann teilnehmen (außer EIM-Mitarbeiter).
2. Vollständige, übersichtliche und lesbare Kontaktdaten müssen unter www.elektor.de/video-wettbewerb hinterlegt werden.
3. Videos müssen von einem Elektor-Projekt oder einem Elektor-Artikel handeln.
4. Videos müssen in englischer, französischer, deutscher oder niederländischer Sprache sein.
5. Videos müssen über WeTransfer eingereicht werden (https://wetransfer.com).
6. Videos müssen vor dem 15. November 2017, 18 Uhr MEZ, eintreffen (die WeTransfer Benachrichtigungs-E-Mail dient als Eingangsstempel).
7. Die Gewinner werden am 22. Dezember 2017 bekanntgegeben.
8. Die Videos dürfen nicht länger als 5 Minuten sein.
9. Videos müssen mit VLC Media Player Version 2.2.6 oder höher abspielbar sein (www.videolan.org).
10. Videos dürfen keine beleidigenden Inhalte enthalten.
11. Videos müssen 100 % urheberrechtsfrei sein.
12. Viel Spaß!




Die genauen Teilnahmebedingungen finden Sie auf

**www.elektormagazine.de/video-wettbewerb**

### Dritter Preis

USB-Digitalmikroskop Andonstar ADSM201
(im Wert von 200 €)

### Extra-Preise

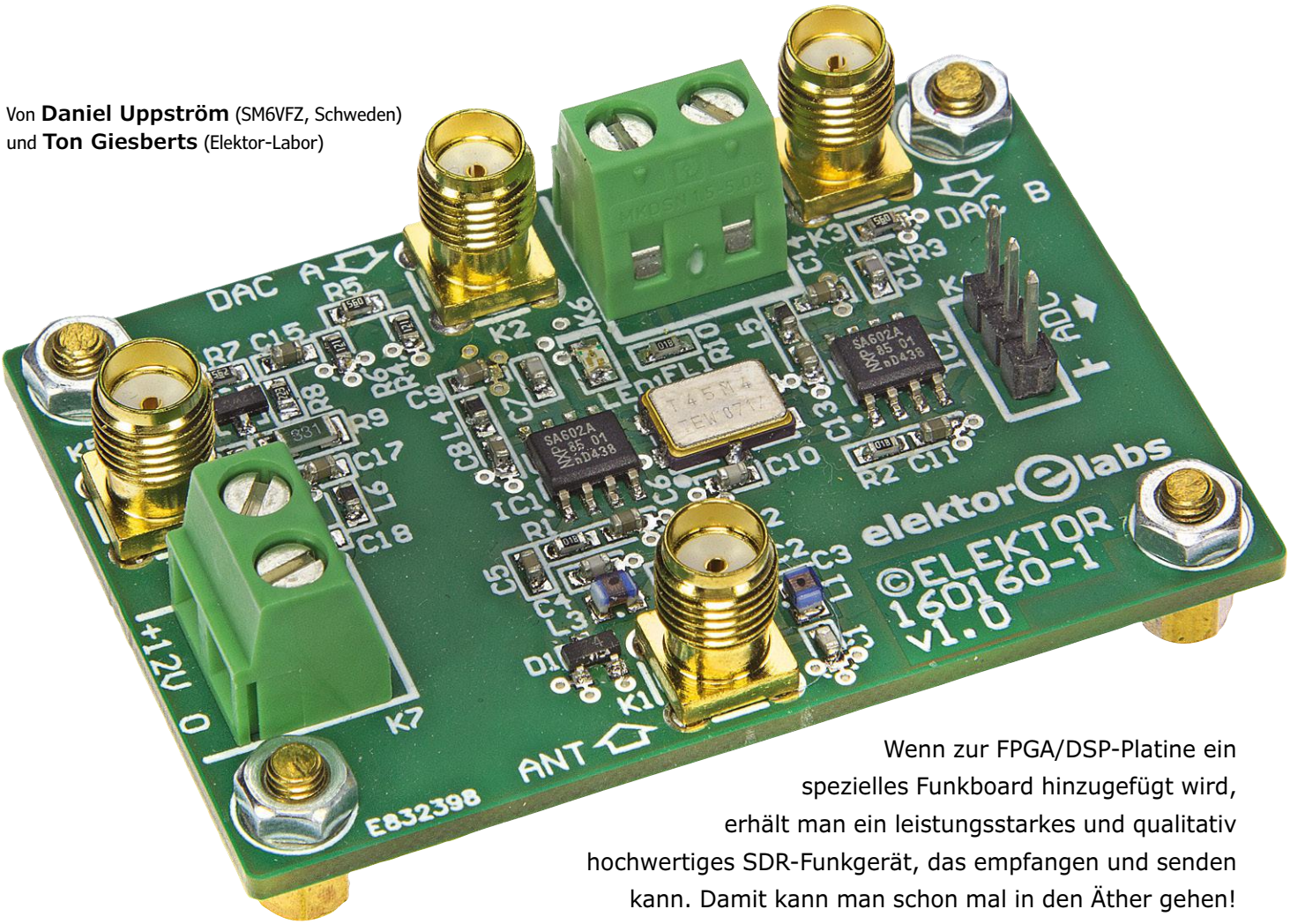
- Sunfounder Smart Video Car für Raspberry Pi
- Raspberry Pi 3 Starter Kit Deluxe
- USB-Digitalmikroskop Andonstar V160
- Kameramodul V2 für Raspberry Pi

02:05 / 04:30

# FPGA/DSP-Board für Schmalband-SDR

## Teil 2: Sender und HF-Frontend

Von **Daniel Uppström** (SM6VFZ, Schweden)
und **Ton Giesberts** (Elektor-Labor)




Wenn zur FPGA/DSP-Platine ein spezielles Funkboard hinzugefügt wird, erhält man ein leistungsstarkes und qualitativ hochwertiges SDR-Funkgerät, das empfangen und senden kann. Damit kann man schon mal in den Äther gehen!

### Eigenschaften

- FPGA + DSP + Audio CODEC
- Schmalband-SDR-Plattform
- Einsetzbar für Funkbetrieb auf nahezu allen Frequenzen zwischen 0 und vielen GHz
- Nach dem Überlagerungsprinzip
- Niedrige zweite ZF höher als 0 Hz
- Kompatibel zu Receive/Transmit-Datenverkehr (RX/TX)
- Weaver SSB-Modulator/Demodulator

Bevor wir uns daran machen, winzige Kondensatoren, Widerstände, Quarzfilter und Ähnliches auf eine Platine zu löten, beenden wir zunächst die Beschreibung der FPGA/DSP-Platine [2]. Im ersten Teil wurden der Empfang und die Dekodierung von Funksignalen zwar ziemlich detailliert behandelt, aber es bleibt noch einiges darüber zu berichten, wie das System im Sendemodus arbeitet. Um Ihnen so viele Hintergrundinformationen wie möglich zu vermitteln, lassen Sie uns einen Blick auf den Sendemodus des FPGA/DSP-Boards werfen. Vielleicht sollten Sie, um das Gedächtnis aufzufrischen, den ersten Artikel nochmals überfliegen und

das Elektor-Heft an die Seite legen, so dass alle (Block-)Schaltpläne „griffbereit“ vor Ihnen liegen.

### Digitale Signalverarbeitung – die Übertragung

Wenn das Push-to-Talk-Signal (PTT) auf dem Header K7 auf logisch low (siehe Teil 1 [2], Bild 2) gezogen wird, schaltet das FPGA in den Sendebetrieb. Der DDS-Block im FPGA kann dann über den DAC einen CW-Träger erzeugen, indem das KEY-Signal auf low gelegt wird.

Die Weaver-Blöcke sind so implementiert, dass sie quasi „rückwärts“ als SSB-Modulator arbeiten können. Wenn das untere oder obere Seitenband gewählt ist und man PTT auf low legt, wird das Audiosignal am Mikrofoneingang abgetastet, komprimiert, tiefpassgefiltert und dann dem Modulator zugeführt. Dies funktioniert sehr gut, erfordert aber eine externe Filterung, da die niedrigen Sampleraten (im kS/s-Bereich) in der Audioabteilung und im Modulator auf die hohe 120-MS/s-Abtastung, wie sie vom DAC verwendet wird, hochgerechnet werden müssen.

Die beiden Spektren in **Bild 1** zeigen 20-m-Sendesignale, wie sie im 14-MHz-Amateurfunkband erzeugt wurden. Beim Graphen links handelt es sich um ein direkt vom DDS erzeugtes CW-Signal. Die diskreten Rauschkomponenten, Nebenwirkungen der Taktsignale im FPGA und der Harmonischen oder „spurs“ um den Träger herum liegen auf ungefähr -70 dBc, das heißt, ihre Amplitude ist -70 dB niedriger als der Träger. Dieses Signal kann problemlos auf 5...10 W verstärkt und übertragen werden, ohne dass man gravierende Störungen riskieren würde.

Das Spektrum auf der rechten Seite in Bild 1 zeigt das Ausgangssignal des Weaver-SSB-Modulators, wenn er ein Audiosignal verarbeitet. Hier sind die Störungen durch die internen Taktsignale viel stärker, außerdem sind auch von der Modulation verursachte starke Alias-Produkte sichtbar. Die unerwünschten Signalkomponenten liegen mit -30 dBc viel höher und sollten nicht einfach verstärkt und übertragen werden: Eine Filterung ist erforderlich.

Abgesehen von den Stör- und Alias-Produkten aus dem Modulator gibt es noch weitere Einschränkungen. Die nicht modulierten Signale, die vom FPGA über den DAC erzeugt werden, sind leider nicht so sauber, wie man es sich wünscht. Dies hat vor allem zwei Gründe:

- Der Takt für den DAC von zunächst 240 MHz – er wird dann durch zwei geteilt - wird von der internen PLL-Schaltung des FPGA erzeugt. Dieses PLL-Master-Taktsignal ist nicht ideal in Bezug auf Phasenrauschen. Es weist einen gewissen Jitter auf, ist also kein perfektes Rechtecksignal mit einer genauen Frequenz und Phase.
- Es gibt auch diskrete Störfrequenzkomponenten am Ausgang. Sie entstehen zum Beispiel durch die unerwünschte Kopplung verschiedener Takte oder Quantisierungseffekte in den DDS-Blöcken und Lookup-Tabellen aufgrund von Wortgrößenbeschränkungen in den Berechnungsschritten, der Speicher- und DAC-Auflösung. Einige dieser parasitären Signale sind im unmodulierten Spektrum links zu sehen.

Alle diese Effekte beschränken die maximale Ausgangsleistung, die im Sendebetrieb möglich ist, wenn man Vorschriften des Amateurfunks respek-

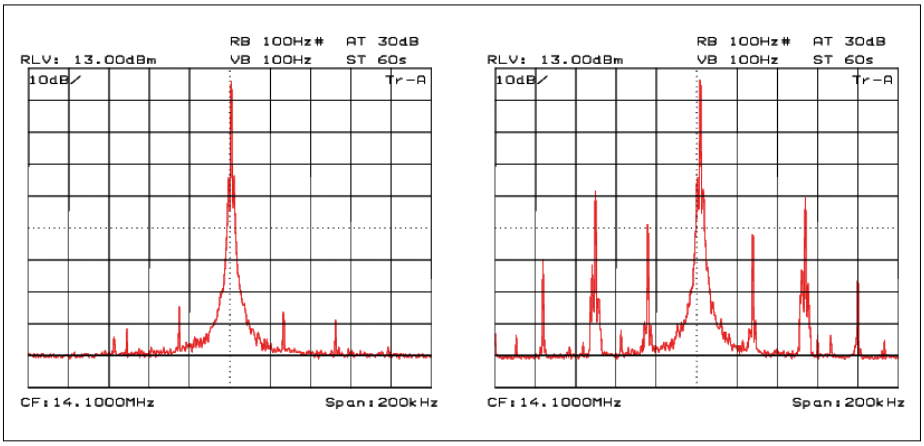



Bild 1. Zwei Spektren eines CW-Signals, das vom FPGA-DSP-Board im Sendebetrieb erzeugt wird (links ohne Modulation, rechts mit Modulation). Die Mittenfrequenz beträgt 14,1 MHz. Die horizontale Auflösung beträgt 200 kHz/div, die vertikale 10 dB/div.

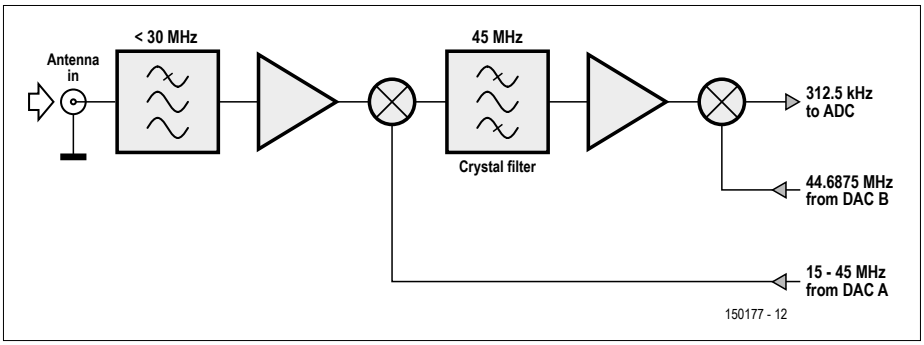



Bild 2. Blockschaltung einer Funk-Erweiterungsplatine.

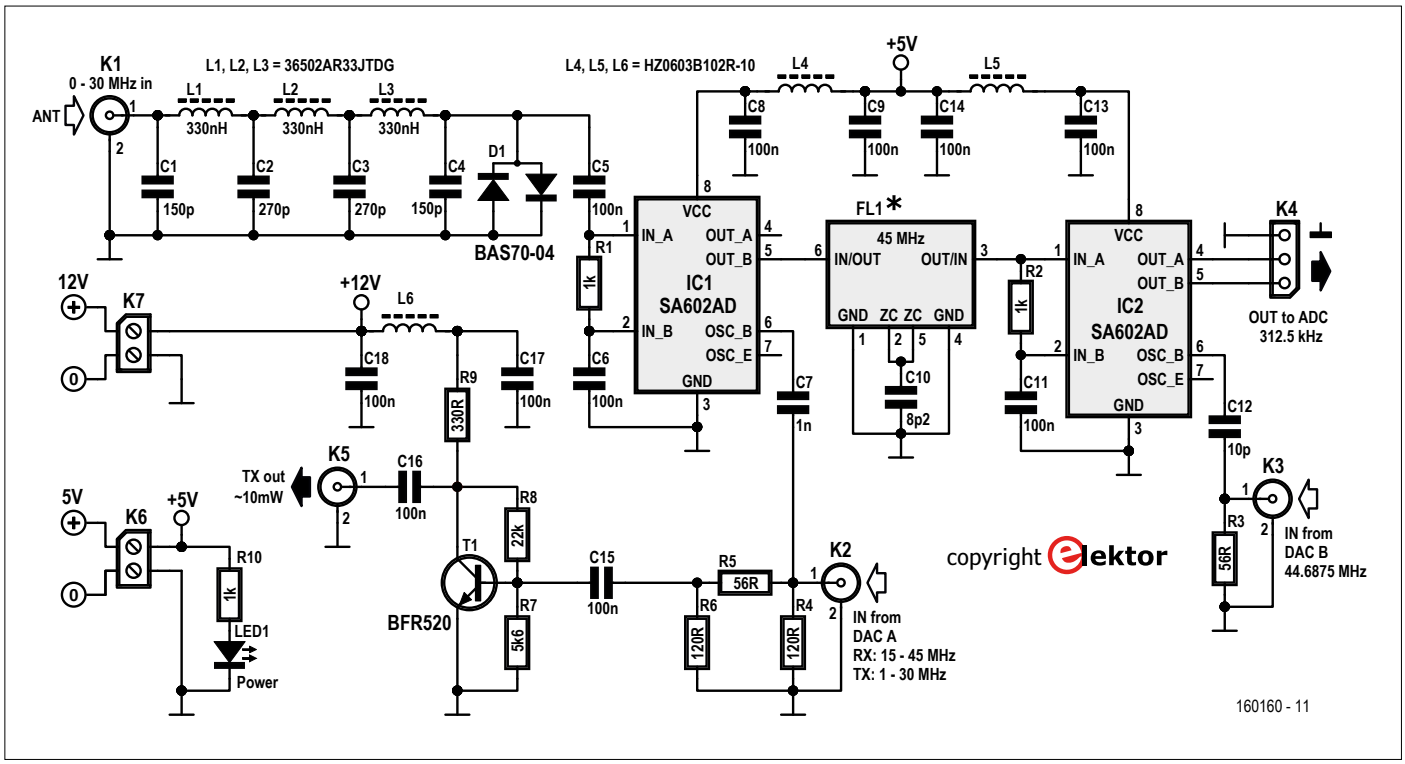



Bild 3. Die Schaltung einer Funkplatine, die für den Empfang und das Senden von Frequenzen bis zu 30 MHz geeignet ist.

tieren und andere Funkdienste nicht mit Interferenzen belästigen möchte. Die Störungen könnten auch einen Einfluss auf Empfänger haben und deren Selektivität und Dynamikbereich einschränken. Wie man solche Probleme dennoch lösen kann, bleibt zu diskutieren.

## Eine einfache Funkplatine

Wir betrachten nun einen einfachen Kurzwellenempfänger (das heißt, bis hinunter zu 30 MHz), dessen Blockschaltung in **Bild 2** zu sehen ist.

Das Antennensignal wird tiefpassgefiltert und verstärkt. Durch Mischen mit dem Signal des lokalen Oszillators (LO) von DAC A wird die relevante Frequenz auf 45 MHz gebracht, bevor sie durch ein Quarzfilter geleitet wird. Dieses Zwischenfrequenzsignal (IF) wird vor einem zweiten Mischer verstärkt, wo es dann mit einem zweiten LO-Signal von DAC B in eine zweite ZF mit einer Mittenfrequenz von 312,5 kHz umgesetzt wird. Dieses zweite ZF-Signal wird vom ADC digitalisiert und zum FPGA geführt, wo es verarbeitet und letztendlich zu einem Audiosignal demoduliert wird.

Eine solche Funkschaltung arbeitet mit dem FPGA/DSP-Board sehr gut zusammen, wenn es um den Empfang von Signalen von nahezu Null bis etwa 30 MHz geht. Mit einer großen Antenne, die sowohl schwache als auch starke Signale auffängt, wird der Empfänger nicht so leicht übersteuert und verzerrt das Audiosignal nicht. Die Schaltung kann daher für den „ernsthaften" Empfang sporadischer und/oder schwacher Signale verwendet werden.

Der detaillierte Schaltplan ist in **Bild 3** dargestellt. Die Antenne ist an K1 angeschlossen. Ihr Signal gelangt direkt zum Tiefpassfilter, das FM-Rundfunksignale und andere potentielle VHF/UHF-Störer entfernt. Die antiparallel geschalteten Dioden D1 begrenzen sehr starke Signale, die durch das Filter gelangen, und stellen auch einen Gleichspannungspfad für statische Ladungen zur Verfügung, die sich sonst in einem langen und isolierten Antennenkabel aufbauen könnten.

IC1, ein SA602, verstärkt das Signal und mischt es mit dem Signal von DAC A (das an K2 angeschlossen ist). Das Quarzfilter FL1 sorgt für eine scharfe Filterung bei der ersten ZF. FL1 mit einer Mittenfrequenz von 45 MHz ist recht schwierig zu erhalten. Idealerweise sollte die Bandbreite 15 kHz betragen, aber mit 30 kHz klappt es auch ganz gut. 7,5 kHz dagegen ist zu schmalbandig für ein hochqualitatives AM-Signal. Die Impedanzen des SA602 und des Filters passen recht gut zueinander, so dass keine zusätzlichen Komponenten zur Impedanzanpassung nötig sind. Ein zweiter SA602 (IC2) mischt das Ausgangssignal von FL1 mit dem vom DAC B erzeugten Signal (via K3), um die zweite ZF als Differenzsignal zu erzeugen, das dann über K4 dem ADC auf der FPGA-Platine zugeführt wird.

## Übertragung

Für einen lizenzierten Funkamateur ist es sicher kein Problem, ein paar Komponenten zu dieser einfachen Funkplatine hinzuzufügen, damit das System als Sender verwendet werden kann. Im Sendemodus erzeugt der DAC-Kanal A einen CW-Träger direkt im 0...30-MHz-Bereich, der von T1 bis zu etwa 10 mW verstärkt wird. Dieses Signal an K5 kann dann einfach mit einem externen Leistungsverstärker auf ein paar Watt angehoben und mit einer abgestimmten Antenne verbunden werden. Schon können Sie mit anderen, weit entfernten Stationen rund um den Globus kommunizieren! Über die maximale Leistung (für CW-Modus/Träger) kann man streiten, aber der Autor würde, wie oben besprochen, zugunsten eines saube-

## STÜCKLISTE

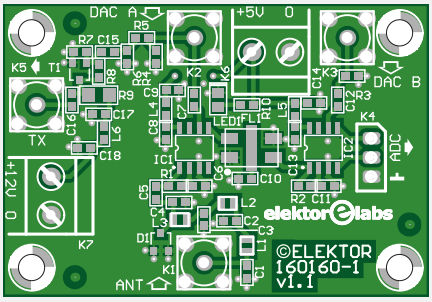
**Widerstände:**
alle 1 %, 100 mW, 0603
R1,R2,R10 = 1 k
R3,R5 = 56 Ω
R4,R6 = 120 Ω
R7 = 5k6
R8 = 22 k
R9 = 330 Ω, 1 %, 250 mW, 1206

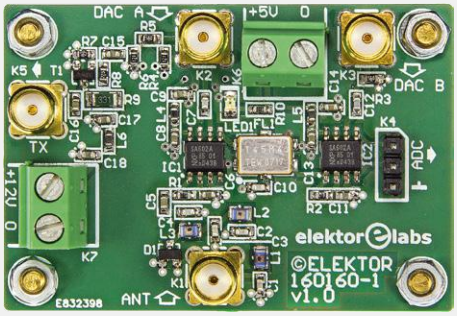
**Kondensatoren:**
alle 0603
C1,C4 = 150 p
C2,C3 = 270 p
C5,C6,C8,C9,C11,C13-C18 = 100 n
C7 = 1 n
C10 = 8p2, ±0.25 p
C12 = 10 p

**Induktivitäten:**
L1,L2,L3 = 330 n, 5 %, 310 mA, $f_{res}$=60 MHz (36502AR33JTDG, TE Connectivity)
L4,L5,L6 = 1000 Ω@100 MHz, 200 mA (HZ0603B102R-10, Laird Technologies)

**Halbleiter:**
D1 = BAS70-04
LED1 = grün, 0805
T1 = BFR520
IC1,IC2 = SA602AD, SO-8

**Außerdem:**
K1,K2,K3,K5 = SMA-Buchse, gerade, 50Ω
K4 = 1x3-polige Stiftleiste, Raster 0,1", gerade
K6,K7 = 2-polige Platinenanschlussklemme, Raster 5 mm
FL1 = Quarzfilter 45 MHz, 1 kΩ, BW=15 kHz (siehe Text), z.B. Golledge-Quarzfilter GSF-75 45R15B1 45.0 MHz
Platine 160160-1




Anzeige

NEU

pico Technology

THE PICOSCOPE 4444

SEE THE DIFFERENCE

A NEW STANDARD IN DIFFERENTIAL MEASUREMENT

• 20 MHz bandwidth, 12 - 14 bit resolution
• 4 fully differential inputs
• 1000 V CAT III probes
• Low voltage probes and current clamps
• 256 MS deep memory
• Measures non-grounded (floating) signals
• Rejects common-mode noise

Für weitere Informationen besuchen Sie www.picotech.com/PS527

Email: Sales@picotech.com ,Irrtümer und Druckfehler vorbehalten.
Für aktuelle Preise kontaktieren Sie bitte vor der Bestellung Pico Technology.

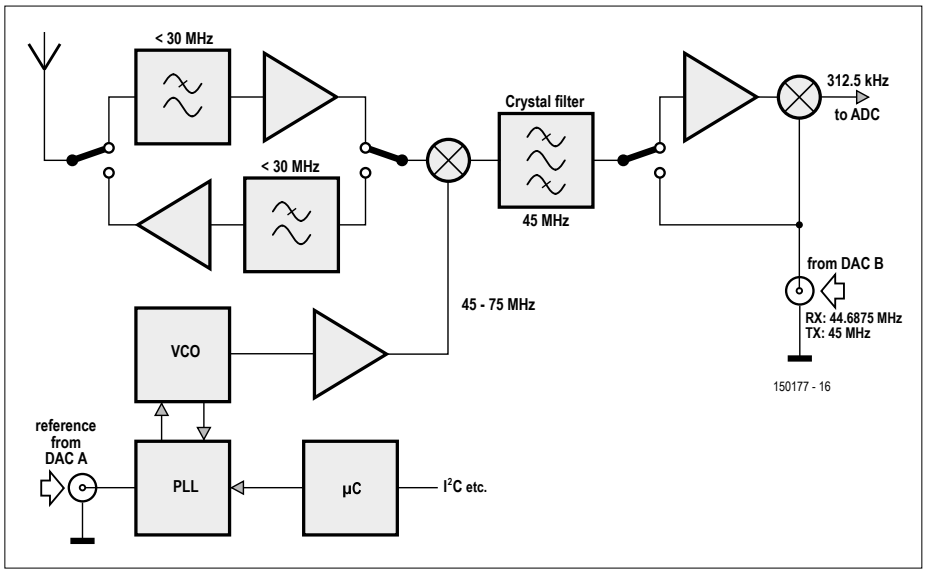



Bild 4. Blockschaltung eines Transceivers für die SSB-Sprachübertragung.

ren Spektrums nicht mehr als 5...10 W auf Kurzwelle empfehlen. Eine solche Leistung ist offensichtlich geringer als die üblichen 100 W handelsüblicher Kurzwellen-Transceiver, die Leistung ermöglicht aber immer noch eine weltweite Kommunikation, wenn die Bedingungen günstig sind, und tagsüber einen interkontinentalen Betrieb zum Beispiel im 40-m-Amateurband (7 MHz).

## Geringes Rauschen, mehr Sendeleistung, VHF und höher

Die hier vorgestellte Funkplatine eignet sich gut für den Empfang und eine CW-Übertragung mit niedriger Leistung und ist deshalb ein guter und einfacher Ausgangspunkt. Ausgangspunkt ja, aber wenn es für ernsthaften Amateurfunk darum geht, einen leistungsstarken Sender/Empfänger (Transceiver) zu bauen, der mit handelsüblichen Geräten vergleichbar oder sogar besser ist und der ein sauberes Sende-Spektrum erreicht, damit auch eine SSB-Übertragung möglich ist, muss doch eine höher entwickelte Funktopologie gewählt werden.

Das Hinzufügen einer Filterbank, die alles außerhalb des abgestimmten Bandes wegfiltert, ist sicherlich verlockend. Es würde helfen, falsche Signale (spurs) und Harmonische loszuwerden, aber das Phasenrauschen und der Jitter rund um den Träger könnten nicht so leicht eliminiert werden. Hier stößt dieser scheinbar einfache Ansatz an seine Grenzen. Es ist auch unwahrscheinlich, dass Alias-Produkte des Modulators weggefiltert werden können, wenn das Filter nicht sehr steil ist. Und dies wäre ziemlich unpraktisch, da es stets auf die genaue Betriebsfrequenz abgestimmt werden müsste.

Eine praktikablere Herangehensweise ist es, einen TX-Träger bei der Quarzfilterfrequenz von 45 MHz zu erzeugen, alle Alias-Komponenten und Störfrequenzen herauszufiltern und dann mit einem lokalen Oszillator (LO) im Bereich von 45...75 MHz auf die Kanalfrequenz von 0...30 MHz herunterzumischen. Dieser LO kann von einem spannungsgesteuerten Oszillator (VCO) erzeugt werden, der mit einem Referenzsignal von einem der DAC-Kanäle phasenverriegelt ist.

Dieser Ansatz ermöglicht eine gute SSB-Sprachübertragung. **Bild 4** zeigt das Blockdiagramm eines solchen Transceivers.

Für höhere Frequenzen, etwa für das 144-MHz-Amateurband, könnte eine ähnliche Topologie verwendet werden, aber es ist dann besser, die erste ZF auf 21,4 MHz zu legen. Für diese Frequenz sind steilerflankige Quarzfilter erhältlich. Der VCO würde dann bei etwa 165 MHz arbeiten.

In diesem 2-m-Band (144 MHz) gibt es manchmal viele starke Signale von lokalen Funkamateuren und Sendern außerhalb des Bandes. Dies stellt höhere Anforderungen an die Blöcke des Empfängers. Hier muss ein guter Mischer und ein möglichst steilflankiges Quarzfilter eingesetzt werden. Es ist auch sehr wichtig, saubere Signale zu senden. Die höhere Antennenverstärkung und die im Vergleich zur Kurzwelle direktere Ausbreitung der Signale können bei den Funkamateuren in der Nähe hohe Signalpegel erzeugen.

## Machen Sie mit!

In diesem Artikel konnten wir (noch) kein nachbaufertiges Funkboard mit voller Leistungsfähigkeit präsentieren. Experimente mit solchen Schaltungen finden gerade statt [4] und ein komplettes 2-Meter-Funkgerät wurde vom Autor auch schon gebaut. Je nach Interesse und Rückmeldung können wir in Zukunft weitere Details veröffentlichen. Einen Transceiver zu bauen ist ein großes Projekt! Funkamateure und experimentierfreudige Elektroniker sind aufgerufen, mit ihren Entwürfen und Anregungen dazu beizutragen.

In einem zukünftigen Artikel wird noch eine Steuerplatine vorgestellt, die sich um die Benutzeroberfläche und die digitale Kommunikation mit dem FPGA kümmert. Damit wird dann der Bau dieses hochflexiblen Funkgeräts abgerundet. ⏮

(160160)

### Weblinks

[1] www.elektormagazine.de/160160
[2] www.elektormagazine.de/150177
[3] www.elektormagazine.com/labs/fpga-dsp-radio-for-narrow-band-communications-150177-i
[4] Blog des Autors: http://sm6vfz.wordpress.com

EINKAUFSZETTEL
→160160-1
Leerplatine für das Funk-Board

# iPhone/iPad-Anwendungen einfach entwickeln

## Externe Elektronik ansteuern über WLAN und BLE

Von **Hans Oppermann** (D)

Wer Anwendungen für ein iPhone oder iPad programmieren möchte, muss einige Hürden meistern. Gerade für Elektroniker, die oft nur eine kleine App zur Steuerung von externer Hardware benötigen, empfehlen sich dagegen einfache Lösungen. Hier kommt techBASIC ins Spiel. Die Plattform erlaubt es, eigene iPhone/iPad-Applikationen direkt auf dem mobilen Gerät zu erstellen; Bibliotheken mit großem Funktionsumfang werden mitgeliefert. In diesem Artikel zeigen wir als Beispiel eine Anwendung, die mit der Außenwelt über Bluetooth Low Energy kommuniziert.

Zum Erstellen von eigenen Apps stellt Apple die kostenlose Entwicklungsumgebung Xcode zur Verfügung. Als Programmiersprache kommt in der Regel Swift zum Einsatz. Die aktuelle Version 8 setzt einen Mac-Rechner mit aktuellem Betriebssystem voraus. Nach der Fertigstellung der App kann diese von Xcode direkt auf das mobile Gerät übertragen werden.

## Die techBASIC-App

Gerade wenn man mit einem Smartphone nur etwas steuern und mit überschaubarem Aufwand eine kleine Benutzeroberfläche (inklusive Interface zu einer eigenen Schaltung) realisieren möchte, ist das Einarbeiten in Xcode sicherlich zu aufwendig. Die Firma Byte Works bietet mit der App „techBASIC" [1] eine interessante Alternative.

Die gesamte Entwicklungsumgebung (IDE) präsentiert sich in einer einzelnen App, die als Demo (techSampler) und als normale Version zum Preis von 16,99 € aus dem App-Store [2] geladen werden kann. Die Demo-Version erlaubt nur das Ausführen der vorhandenen Demoprogramme, das heißt die Erstellung neuer Programme beziehungsweise die Änderung von Programmen sind nicht möglich. Nach dem Start der App werden alle vorhandenen Programme (Demos) eines Verzeichnisses angezeigt. Aus diesem Verzeichnis heraus können die Programme gestartet oder bearbeitet werden. Die fertigen Programme lassen sich auch zu Apps konvertieren. Dazu ist das Programm techBASIC App Builder zum Preis von 49 $ und ein Mac erforderlich, auf dem die App generiert wird. Für die meisten Elektroniker wird dies aber wohl überflüssig und zu aufwendig sein.
Die Programmentwicklung auf dem iPhone macht sicherlich wegen des kleinen Displays wenig Sinn. Auf einem iPad mit externer Tastatur kann man aber sehr schön Programme entwickeln, sowohl für das iPad als auch für das iPhone. Die App muss selbstverständlich nur einmal gekauft werden.

**Listing 1. Verarbeitung von Benutzereingaben am Beispiel eines WLAN-Clients, der mit einem ESP8266 kommuniziert.**

```
! Handle pressed button
SUB touchUpInside (ctrl AS Button, time AS DOUBLE)
   http_string = "http://192.168.254.70/"

! Heizung EIN
    IF ctrl = Tag THEN
        http_string = http_string & "gpio/1"
        proc_http
    END IF
! Heizung AUS
    IF ctrl = Nacht THEN
        http_string = http_string & "gpio/0"
        proc_http
    END IF
! Lese Status
    IF ctrl = Status THEN
        http_string = http_string & "gpio/read"
        proc_http
    END IF
! Lese Vorlauftemperatur
    IF ctrl = VL_Temp THEN
        http_string = http_string & "temp"
        proc_http
    END IF
    IF ctrl = quit THEN
        STOP
    END IF
END SUB
```

## Was bietet techBASIC?

Der Name lässt sicherlich schon vermuten, dass hier sehr viele Ähnlichkeiten zu der alten Programmiersprache BASIC bestehen. Der Funktionsumfang ist aber überhaupt nicht mit dem ursprünglichen BASIC vergleichbar. Eine vollständige Beschreibung enthält das techBASIC Reference Manual [3]. Außer den üblichen Mathematik- und Zeichenkettenfunktionen (wie zum Beispiel LEFT$) hat techBASIC verschiedene Funktionen zur Event-Bearbeitung zu bieten. Hiermit kann beispielsweise auf das Drücken eines Display-Buttons reagiert werden.
Diverse Funktionen entfallen auf Bluetooth Low Energy (BLE) und WLAN – sie ermöglichen die Kommunikation mit der Außenwelt. BLE und WLAN sind auch gleichzeitig die beiden einzigen Kommunikationsprotokolle, die von Apple zugelassen sind, nicht aber USB oder Bluetooth 2.x. Die beiden genannten Funkprotokolle sind jedoch sicherlich auch die interessantesten Kommunikationswege, um beispielsweise externe Sensoren im Bereich des IoT oder der Hausautomatisierung abzufragen.
WLAN wird durch die Funktion/Class `Comm` unterstützt. Hiermit lässt sich auf sehr einfache Weise ein HTTP-Client auf dem iPhone/iPad realisieren. Als Beispiel habe ich Code (OnOffTemp.txt) zum Steuern eines Heizkessels mit einem ESP8266-HTTP-Server entwickelt. **Listing 1** zeigt den Programmteil zur Verarbeitung der Eingaben aus der Benutzeroberfläche, den gesamten Code kann man von der Elektor-Website herunterladen [4].
Auch zum Auslesen der internen Gerätesensoren (Beispiel Beschleunigung und geographische Koordinaten) stellt techBASIC entsprechende Funktionen bereit. Die grafischen Funktionen, mit denen komplexe Kurven und grafische Darstellungen erzeugt werden können, möchte ich nur am Rande erwähnen.
Viel wichtiger (sozusagen das Tüpfelchen auf dem i(Phone)) erscheinen mir die umfangreichen Funktionen zur Erstellung von grafischen Benutzeroberflächen (GUI). Mit wenigen Befehlen sind hier die eindrucksvollen, in vielen kommerziellen Apps verwendeten Widgets (Buttons, DatePicker, Slider, …) zu erstellen. Auf der Internetseite des techBASIC-Herstellers gibt es eine Demo, die alle Widgets auf einem iPhone/iPad-Display vereint.

## Beispiel-Schaltung

Das folgende Beispiel demonstriert das Auslesen eines Klimadaten-Sensors mit techBASIC. Dieser erfasst Temperatur, Luftfeuchtigkeit, Luftdruck und UV-Strahlung. Die gemessenen Werte werden mit BLE an ein iPhone/iPad gesendet, auf dem das techBASIC-Programm läuft. Für die BLE-Übertragung wird das Modul BL600 verwendet, das den meisten Elektor-Lesern sicherlich bekannt sein wird. Als Sensoren werden zwei Breakout-Boards (BoB) genutzt, zur Messung der UV-Strahlung ein Adafruit Si1145 und für die anderen drei Größen ein BME280-BoB. Das ganze Gerät wird mit einer Knopfzelle CR2032 betrieben, welche bei mir schon seit rund 10 Monaten ununterbrochen in Betrieb ist.

Die Schaltung ist in **Bild 1** zu sehen. An JP1 wird bei Bedarf ein „USB to UART Converter" zur Programmierung und Debugging des BL600 angeschlossen. JP5 dient zur Einstellung der BL600-Modi (diese sind schon ausführlich in Elektor behandelt worden). In dieser Anwendung muss der Jumper in der Position „AUTORUN" gesetzt sein. Eine kleine Besonderheit stellt der Taster S1 dar. Er ermöglicht in Zusammenarbeit mit der Software den geringen Strombedarf des Gerätes. Das Gerät befindet sich normalerweise im sogenannten „deep sleep modus", der einen sehr niedrigen Strombedarf hat. Erst durch Betätigung des Tasters S1 wird es aufgeweckt, liest zyklisch die Sensoren aus und versendet die Messwerte über Bluetooth. Nach rund zwei Minuten fällt der BL600 wieder zurück in den Schlafmodus. Die BL600-Software ($autorun$.klima.sb) ist bei Elektor downloadbar [4].

## Scannen nach BLE-Geräten

Nun aber wieder zur Gegenseite, dem Empfang der Sensordaten mit techBASIC auf dem iPhone/iPad. Den gesamten Code hier zu beschreiben wäre sicherlich etwas zu aufwendig. Vieles im Code ist selbsterklärend und kann sicherlich nach kurzem Studium verstanden werden. Es soll an dieser Stelle lediglich gezeigt werden, dass mit wenigen Codezeilen eine Verbindung zu einem BLE-Gerät aufgebaut werden kann – siehe hierzu **Listing 2**.

Die Funktion `scanBLE` wird aufgerufen, wenn der Button „Scan" in der GUI gedrückt wird. Ein erfolgreicher Scan wird über den Aufruf der Funktion `BLEDiscoveredPeripheral` gemeldet. Die-

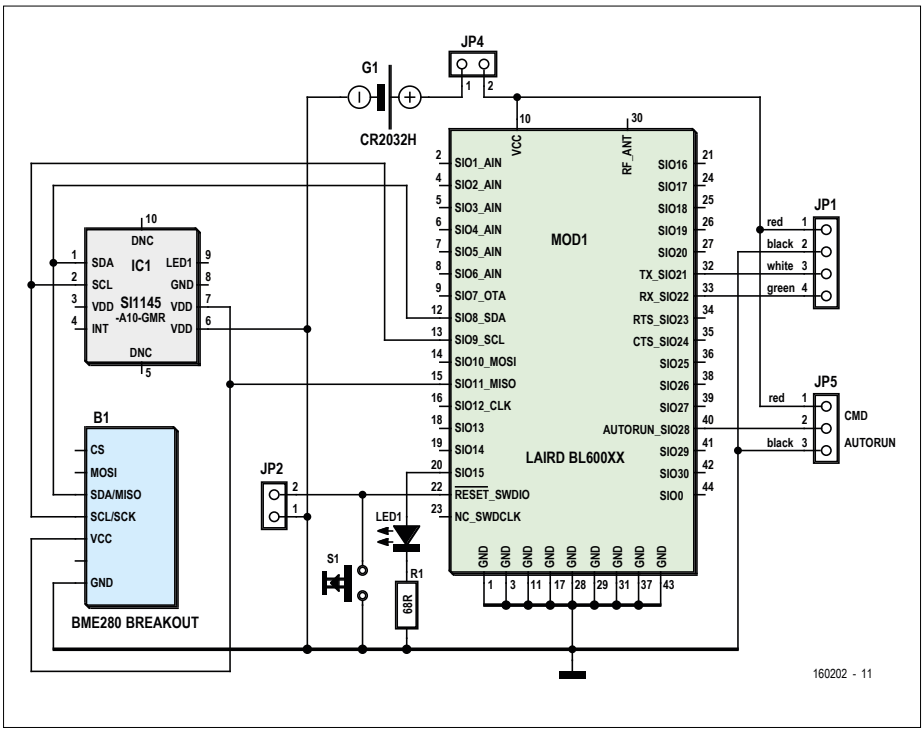



Bild 1. Die Schaltung der externen Elektronik mit den Sensoren und dem BLE-Modul.

**Listing 2. Scannen der BLE-Umgebung bei der Sensor-Anwendung.**

```
!---------------- scan for BLE Device ----------------------------
SUB scanBLE
! Start the BLE service and begin scanning for devices.
    BLE.startBLE
    BLE.startScan(uuid)
END SUB

! Called when a peripheral is found. If it is a LAIRD BL600, we
! initiate a connection to it and stop scanning for peripherals.
!
! Parameters:
! time - The time when the peripheral was discovered.
! peripheral - The peripheral that was discovered.
! services - List of services offered by the device.
! advertisements - Advertisements (information provided by the
! device without the need to read a service/characteristic)
! rssi - Received Signal Strength Indicator
!
SUB BLEDiscoveredPeripheral (time AS DOUBLE, peripheral AS
BLEPeripheral,services() AS STRING, advertisements(,) AS
STRING,rssi)
    PRINT peripheral.bleName
    IF peripheral.bleName = "LAIRD BL600" THEN
        bl600 = peripheral
        BLE.connect(bl600)
        BLE.stopScan
    END IF
END SUB
```

ser Funktionsname und auch die anderen Funktionsnamen, die beim Verbindungsaufbau aufgerufen werden, sind fester Bestandteil von techBASIC. Es müssen also keine Callback-Funktionen (so wie bei vielen anderen Programmiersprachen) selbst definiert werden. Als Nächstes wird die Funktion `BLEPeripheralInfo` aufgerufen. Diese fragt die verfügbaren *Services* auf dem *BLE-Device* ab. Wenn man diese nicht aus der Dokumentation des BLE-Devices entnehmen kann, so hilft ein kleines techBASIC-Programm mit dem Namen „sniffer". Dieses zeigt alle verfügbaren Services des BLE-Devices in Form einer UUID an.
Ist der gewünschte Service (hier *Virtual Serial Port Service*) in `BLEPeripheralInfo` selektiert, so wird die sogenannte *Characteristic* des Services angefordert. Diese wird dann in der Funktion `BLEServiceInfo` empfangen und es werden gleichzeitig mit einem *Notify Request* die Daten des BLE-Devices angefordert. In der Funktion `BLECharacteristicInfo` werden die Daten entgegengenommen und in die entsprechenden Variablen des GUI geschrieben.

## Die Benutzeroberfläche

Die Funktion `showGUI` (**Listing 3**) erzeugt die einzelnen Widgets (hier nur Buttons und Labels) des GUI und zeigt sie an den entsprechenden Koordinaten des iPhone/iPad-Displays an (**Bild 2**). Sie wird einmalig beim Start des Programms aufgerufen.

Zu dem Programm ist noch zu bemerken, dass ich dieses keineswegs von Null an selbst geschrieben habe. Es ist lediglich eine Anpassung und Erweiterung der vielen mit techBASIC gelieferten Beispiele. In diesem Fall diente das Programm „KeyFob" als Vorlage. Das hier beschriebene Programm läuft sowohl auf dem iPhone als auch auf dem iPad. Hierzu ist natürlich eine Anpassung der grafischen Koordinaten an das jeweilige Gerät erforderlich. Der Gerätetyp kann aber im Programm abgefragt werden, so dass nur eine Programmversion für beide Geräte benötigt wird. Das komplette Programm (klima.txt) ist über Elektor abrufbar [4].

## Die Vorläuferanwendung unter Android

Vor ein paar Jahren war es noch notwendig, am *Apple Developer Program* teilzunehmen, um eigene Apps auf das iPhone oder iPad zu übertragen. Die Teilnahme

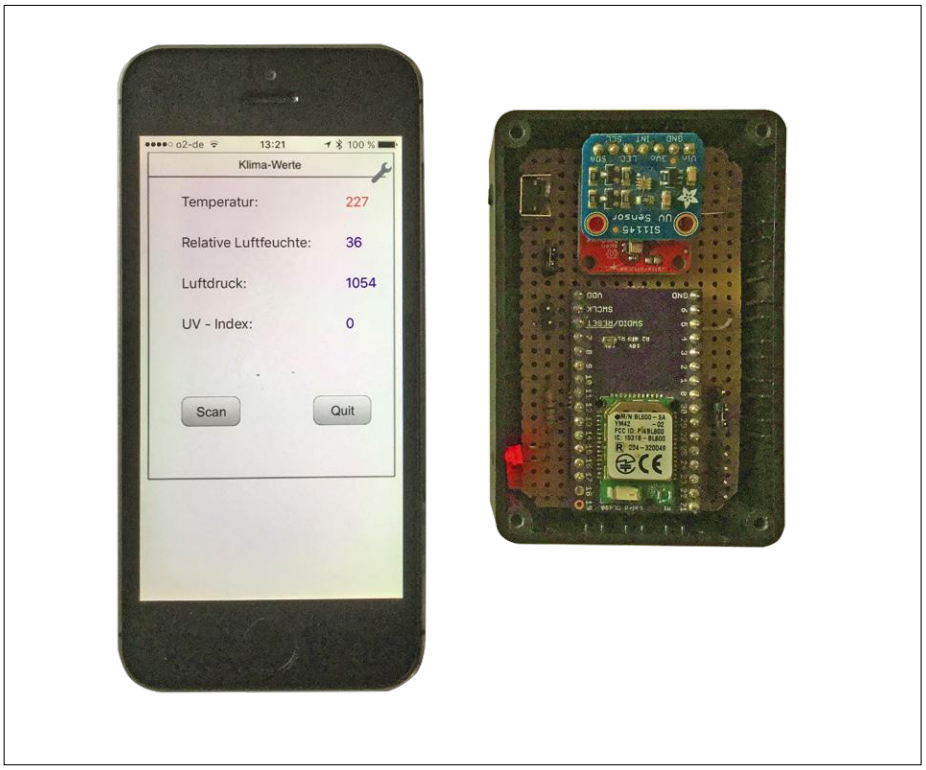



Bild 2. Die Daten werden vom iPhone über BLE empfangen und angezeigt.

**Listing 3. Definition der Benutzeroberfläche der Sensor-Anwendung (Auszug).**

```
!------------------------- showGUI -------------------------------

SUB showGUI
    x_lab = 50
    y_lab = 50
    y_distance = 50

    DIM tempLabel AS Label
    !tempvar = "20"
    tempLabel = Graphics.newLabel(x_lab, y_lab, 100)
    tempLabel.setText("Temperatur: ")
    tempvarLabel = Graphics.newLabel(x_lab+200, y_lab, 100)
    tempvarLabel.setText(tempvar)
    tempvarLabel.setColor(1,0,0)

    y_lab = y_lab + y_distance

    DIM humLabel AS Label
    !humvar = "30"
    humLabel = Graphics.newLabel(x_lab, y_lab, 200)
    humLabel.setText("Relative Luftfeuchte: ")
    humvarLabel = Graphics.newLabel(x_lab+200, y_lab, 100)
    humvarLabel.setColor(0,0,1)

    y_lab = y_lab + y_distance

    DIM airpLabel AS Label
```

kostet jährlich rund 100 $. Aus diesem Grund habe ich mich damals entschlossen, die gleiche App unter Android zu entwickeln (KLIMA.zip). Glücklicherweise gab es zu dem Zeitpunkt die Artikelserie zum „eBoB BL600“ ab März 2015 in Elektor, in der es unter anderem auch um eine Temperaturmessung ging. Auf Basis dieses Programms war es mir möglich, eine entsprechende Android-Applikation zu erstellen. Zugegebenermaßen habe ich dabei auch einiges über Android und die Android-Entwicklungsumgebung „Android Studio“ gelernt. Aber der Aufwand war im Verhältnis zum Ergebnis (Auslesen von ein paar Sensoren und über BLE senden) sehr hoch. Die grafische Oberfläche sieht lediglich ein wenig hübscher aus (**Bild 3**) als bei der techBASIC-Lösung.

## Fazit

Ich bedauere, dass ich techBASIC nicht schon früher entdeckt habe, dann wäre mir der aufwendige Ausflug in die Android-Welt erspart geblieben. In Zukunft werde ich also nicht mehr mit zwei Handys umherlaufen, sondern kann all meine Bluetooth-Anwendungen auf das iPhone übertragen. Allerdings erfordert dies auch noch eine Umrüstung der in meinen Projekten eingesetzten Bluetooth-Chips auf Bluetooth 4.0 (BLE). Aber das dürfte nicht allzu aufwendig sein.

Für jeden, der sich intensiver mit dem Thema beschäftigen möchte, kann ich das Buch „Building iPhone and iPad Electronic Projects, Real-World Arduino, Sensor and Bluetooth Low Energy Apps in techBASIC“ [5] empfehlen. ⏮

(160202)

### Weblinks


[1] www.byteworks.us/Byte_Works/techBASIC.html

[2] https://itunes.apple.com/de/app/techbasic/id470781862?mt=8

[3] www.byteworks.us/Byte_Works/Documentation_files/techBASIC%20Manual%203.3.1.pdf

[4] www.elektormagazine.de/160202

[5] http://shop.oreilly.com/product/0636920029281.do?sortby=publicationDate


```
    !airpvar = "40"
    airpLabel = Graphics.newLabel(x_lab, y_lab, 200)
    airpLabel.setText("Luftdruck: ")
    airparLabel = Graphics.newLabel(x_lab+200, y_lab, 100)
    airparLabel.setColor(0,0,1)

    y_lab = y_lab + y_distance

    DIM uviLabel AS Label
    !uvivar = "50"
    uviLabel = Graphics.newLabel(x_lab, y_lab, 200)
    uviLabel.setText("UV - Index: ")
    uviparLabel = Graphics.newLabel(x_lab+200, y_lab, 100)
    uviparLabel.setColor(0,0,1)

    !Create a scan button.
    scan = Graphics.newButton(but1_pos ,300)
    scan.setTitle("Scan")
    scan.setBackgroundColor(1, 1, 1)
    scan.setGradientColor(0.7, 0.7, 0.7)

    !Create a quit button.
    quit = Graphics.newButton(but2_pos ,300)
    quit.setTitle("Quit")
    quit.setBackgroundColor(1, 1, 1)
    quit.setGradientColor(0.7, 0.7, 0.7)

END SUB
```

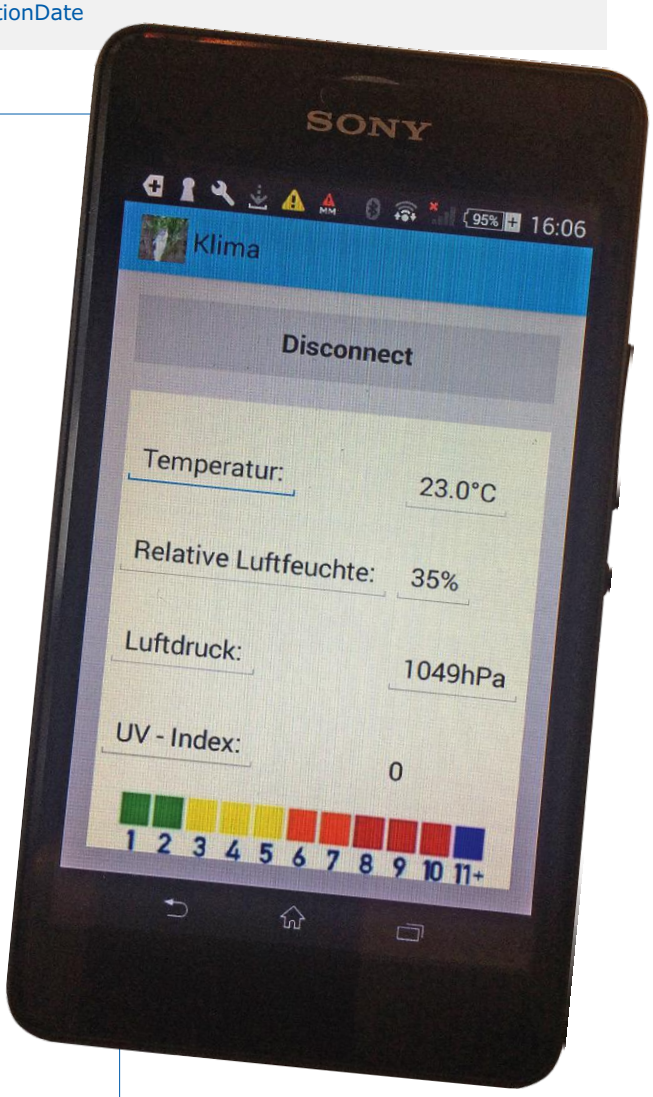
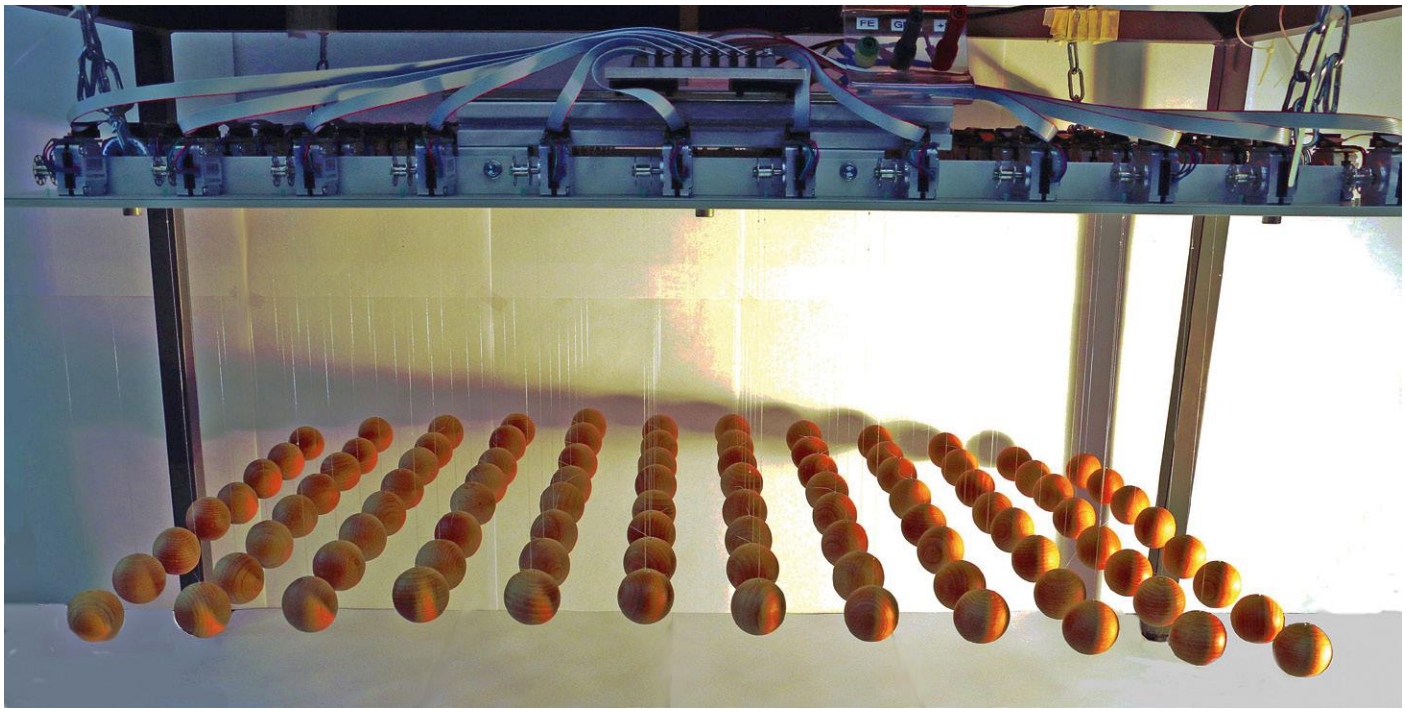



Bild 3. Zum Vergleich: Die Beispielanwendung unter Android.

# Kinetische Skulptur

## Treffen sich ein E-Techniker und ein Künstler...



Von **Andreas Lambers** und **Michael Bragard**

Die Verbindung der Welten dressierter Elektronen und grenzenloser Kreativität bietet ein großes Potential; zum Beispiel bei modernen Skulpturen, deren Form sich durch Motoren verändern kann. An der FH Aachen wurde ein solches Projekt verwirklicht: Eine Matrix aus Holzkugeln kann Piktogramme anzeigen, aber auch mathematische Funktionen visualisieren. In diesem Artikel beschreiben wir die clevere Ansteuerung der Motoren.

Ein Treffen von Künstler und Ingenieur hat sich wohl direkt hinter dem Kassenbereich des Münchener BMW-Museums ereignet, wo eine großformatige „kinetische Skulptur" den Besucher empfängt. Diese Skulptur besteht aus 714 silberfarbenen Kugeln, die einzeln an Stahlseilen durch Motoren von der Decke heruntergelassen werden [1]. Die Bewegungen der Kugeln sollen den Designprozess bei der Formfindung von Karosserieteilen darstellen – in jedem Fall ist das Ganze ein gelungener Blickfang.

Dieser Beitrag beschreibt, wie eine solche Visualisierungsidee den Weg in Ihr heimisches Wohnzimmer finden kann. Entstanden ist das hier beschriebene Projekt im Rahmen einer Bachelorarbeit an der Fachhochschule Aachen (Fachbereich Elektro- und Informationstechnik).

### Hardware in der Übersicht

Unsere kinetische Skulptur besteht aus einer Matrix von 12x8 Holzkugeln. Im Vergleich zur BMW-Skulptur ist dies zwar nur „Holzklasse", aber dennoch ein prima Ansatz für eigene Designs und Erweiterungen. Jede Kugel hängt an einem kaum sichtbaren Nylonfaden, der, auf der Fadenspule einer Nähmaschine aufgewickelt, an einem Schrittmotor befestigt ist. Die einzelnen Module sind in konstantem Abstand auf einem Blech montiert. Wie **Bild 1** zeigt, sind die Schrittmotoren beziehungsweise die Ansteuerschaltungen für eine Reihe von Kugeln parallel über Flachbandkabel mit einer zentralen Platine, dem Hub, verbunden. Für den ersten Versuch haben wir, wie das Titelbild zeigt, das Blech mit den Schrittmotoren an einem Tischgestell aufgehängt, sodass die Kugeln frei im Raum zu schweben scheinen.

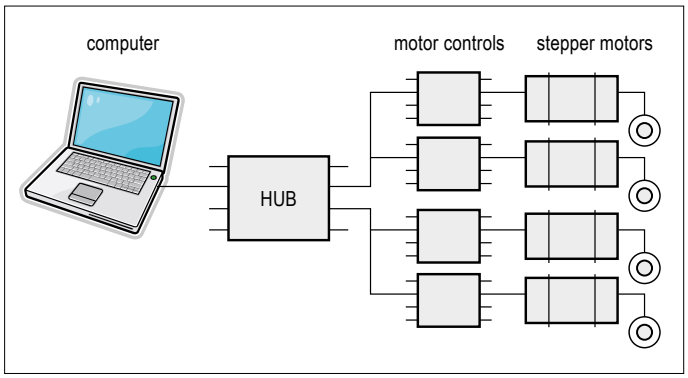



Bild 1. Eine Netzwerktopologie im Format „gemischter Stern“.

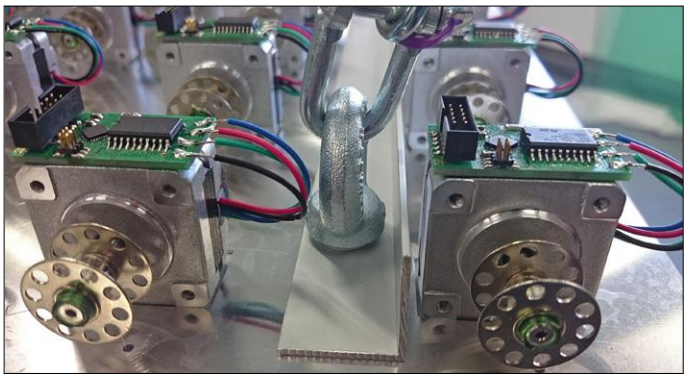


Bild 2. Schrittmotor mit individueller Steuerplatine.

**Die Auswahl der Schrittmotoren**

Schrittmotoren sind aufgrund ihrer vergleichsweise einfachen Ansteuerung und der guten Verfügbarkeit bei geringen Kosten für dieses Vorhaben prädestiniert. Bei der Auswahl der Motoren spielen vor allem das verfügbare Drehmoment, die Leistungsaufnahme und die Befestigung des Motors eine entscheidende Rolle. Für eine einfache Montagemöglichkeit bietet sich die Bauform NEMA 14 an. Nach einiger Recherche fiel die Wahl auf den häufig in Restposten angebotenen Schrittmotortyp PSM35BYG104 (**Bild 2**). Vorversuche zeigten, dass das verfügbare Rastmoment von 0,5 Ncm hinreichend groß ist. Dieser Parameter ist besonders wichtig, da ansonsten alle Kugeln im stromlosen Zustand des Motors absinken würden. In Kombination mit einem Spulendurchmesser von unter einem Zentimeter darf das Gewicht der Kugel etwa 100 g betragen. Der gewählte Motor verfügt über 200 physikalische Schritte, das heißt, er erreicht im Vollschrittbetrieb eine Auflösung von 1,8 °. Dies entspricht bei der gewählten Mechanik einer Positioniergenauigkeit von ungefähr 0,1 mm!

**Dezentrale Motorsteuerung**

Wie in Bild 2 zu sehen, ist auf jedem Schrittmotor eine kleine Steuerplatine befestigt. welche die Informationssignale des zentralen Hubs entgegennimmt und diese mit einer Leistungsendstufe für den Motor aufpeppt.

Die Schaltung der Steuerplatine (**Bild 3**) besteht aus einem Mikrocontroller ATmega88-20MU [2], der über die vier Anschlüsse PC0...PC3 (IN1...IN4) die Kommunikation und die Steuerung der beliebten doppelten Vollbrücke L293DD [3] übernimmt. In diesem IC stecken zwei Treiberpaare, an deren Ausgängen (also in der Mitte der Brücke) jeweils eine Wicklung des bipolaren Schrittmotors angeschlossen wird. Der Mikrocontroller erzeugt eine Schaltsequenz, die den Rotor schrittweise vorwärts oder rückwärts bewegt. Außerdem kann der Mikrocontroller über PD6 und PD7 (EN1, EN2) die Brückentreiber komplett ein- und ausschalten.

Der L293DD wird hier mit 5 V Versorgungsspannung betrieben, ist jedoch auch für eine Zwischenkreisspannung von 12 V (mehr ist mit dem gewählten Motor nicht möglich) einsetzbar. Die Treiberstufe arbeitet mit bis zu 1 kHz Taktrate für die gewünschte Schrittfolge. Bei 5 V ist ein Maximalstrom von 600 mA möglich, der dauerhaft aufrechterhalten werden kann. Für den gewählten Motor ist dies mehr als ausreichend. Der Shunt-Widerstand R2, der mit den ADC-Eingängen des Controllers verbunden ist,

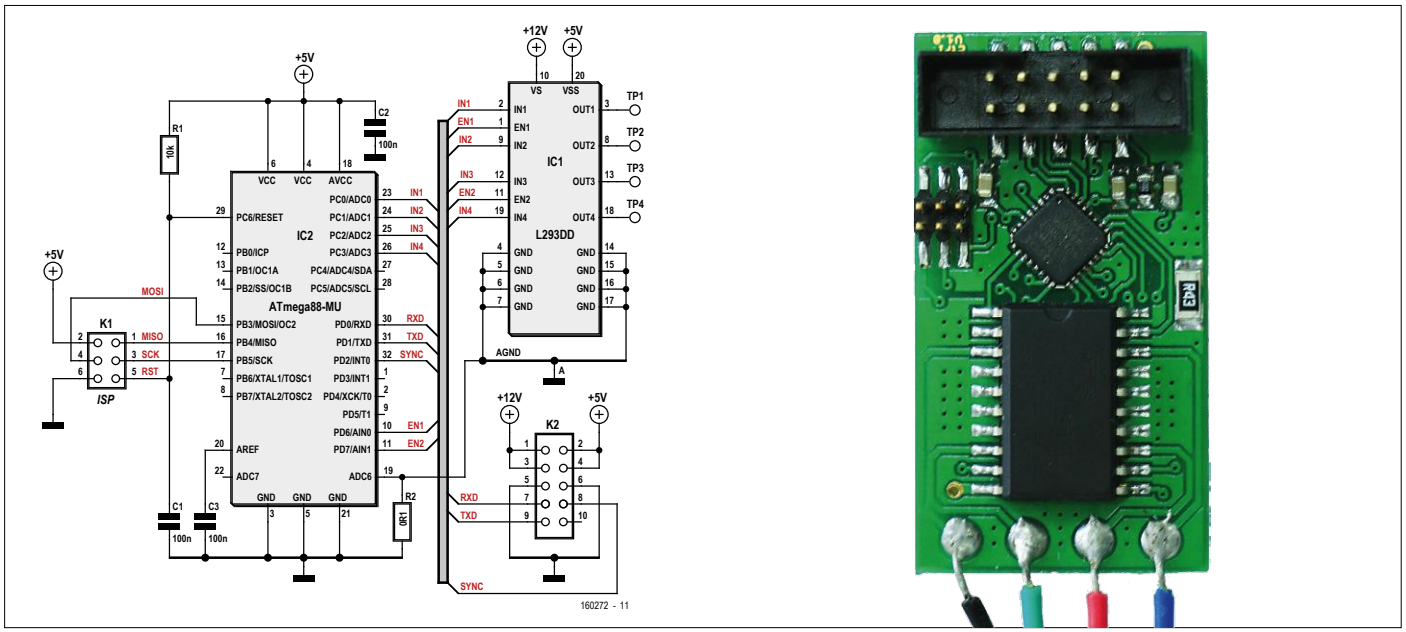



Bild 3. Schaltung und Layout der Schrittmotor-Steuerplatine.

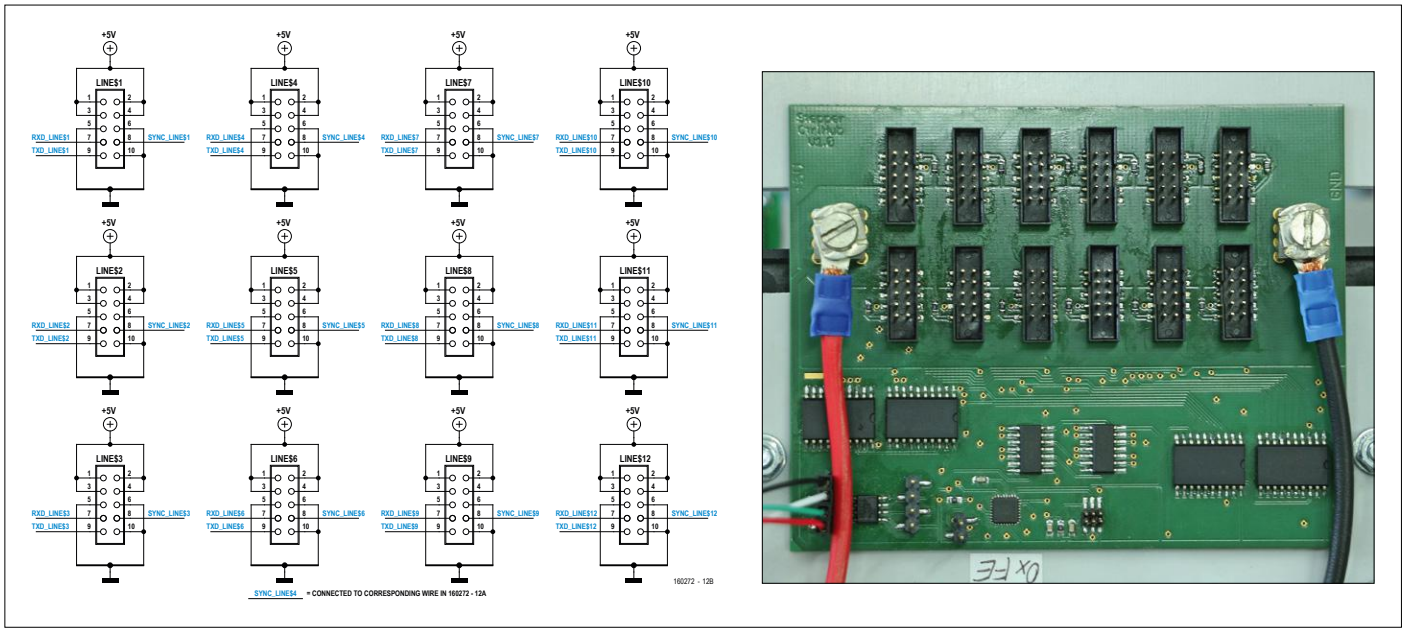



Bild 4. Auf der zweigeteilten Hub-Platine befinden sich oben die Motorstrang-Anschlüsse.

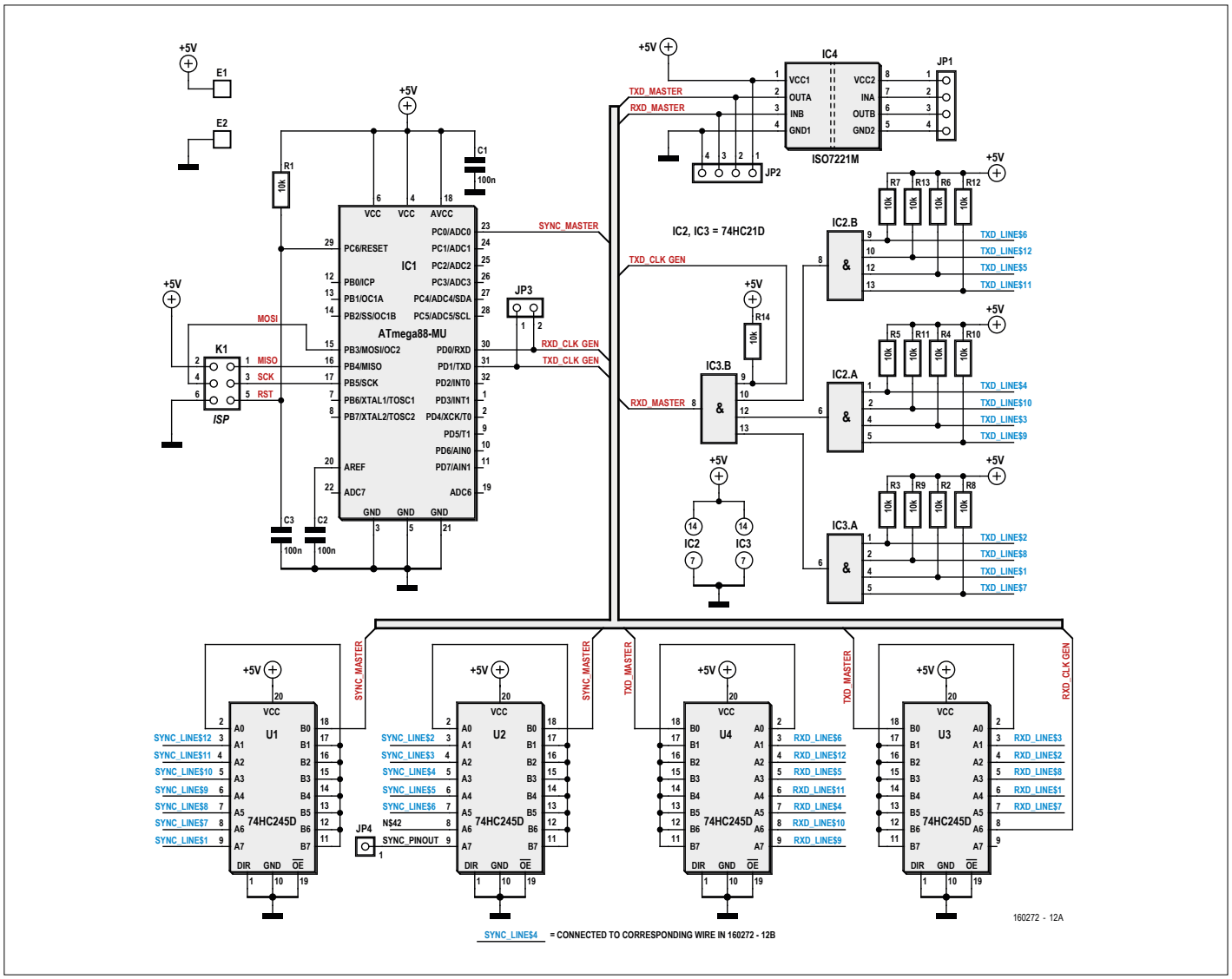



Bild 5. Die Schaltung der Hub-Kommunikationseinheit.

dient der Messung des momentan fließenden Stroms. Der Vergleich der ADC-Eingangsspannung mit der internen Referenz des Controllers führt zu einer guten Abschätzung des Stroms (und bei zu hohen Werten zum Abbruch des Betriebs).
Für die Kommunikation mit dem zentralen Hub sind drei Leitungen vorgesehen. Das vom zentralen Hub stammende Taktsignal SYNC auf Pin 32 (PD2) sorgt dafür, dass alle Motoren synchron laufen. Die für jeden Motor individuelle Kommunikation RxD und TxD mit der zentralen Steuerplatine erfolgt über den UART des Controllers (Pin 30 und Pin 31).
Die restliche Beschaltung der Platine kümmert sich um die Stabilisierung der Versorgungs- und der Referenzspannung; auch ist mit R1/C1 ein Power-on-reset-Glied (PoR) vorhanden. Die kleine 6-polige Stiftleiste ist die Programmierschnittstelle des Controllers und weist die übliche AVR-ISP-Pinbelegung auf.

### Der Hub, die zentrale Kommunikationseinheit

Herzstück der zentralen Kommunikation ist der Hub als Kern des Busses, der die Synchronität der Bewegung der Einzelmotoren garantiert. Gleichzeitig erfolgt auch die Spannungsversorgung der einzelnen Motoren samt Elektronik über den Hub. Aufgrund des topologischen Aufbaus des Netzwerks als gemischte Sternstruktur [4] bildet der Hub einen Flaschenhals hinsichtlich Zuverlässigkeit und Betriebsstabilität.
In **Bild 4** ist zu sehen, dass auf der Platine Leistungs- und Kommunikationsteil klar voneinander getrennt sind. Durch diese räumliche Trennung sind Störungen nahezu ausgeschlossen. Um nicht in einem Wirrwarr von Kabeln zu ersticken, sind die acht Motoren eines Strangs an jeweils nur ein Flachbandkabel angeschlossen. Damit ist der Leistungsteil mit zwölf Stiftverbindern (2x5-polige Stiftleisten), jeder für eine Reihe, recht übersichtlich bedeckt. Rechts und links neben den Verbindern befinden sich die Anschlüsse für ein externes Netzteil. Der Summenstrom aller Motoren liegt im Bereich zwischen 15 A und 20 A. Aus diesem Grund kommen auch spezielle Hochstromklemmen von Würth [5] und auf der Platine Kupfer-Polygone zum Einsatz, um die stromführende Querschnittsfläche zu maximieren. Auch wird der Strom für die Motorenmodule stets über mehrere Stifte der Verbinder geführt.
Auf dem unteren Teil der Hub-Platine befindet sich die Kommunikationsabteilung, deren Schaltung in **Bild 5** zu sehen ist. Sie ist das Bindeglied zwischen dem steuernden PC, der an JP1 angeschlossen wird und die Signale TxD_MASTER und RxD_MASTER empfängt beziehungsweise liefert, und den individuellen Steuerungen der Schrittmotoren. Auch hier regiert ein ATmega88 im MLF32-Gehäuse mit der gleichen Standardbeschaltung wie auf den Motormodulen, doch reicht die Kraft seiner I/O-Pins bei weitem nicht aus, um alle Motormodule mit SYNC-Signalen zu versorgen. Deshalb spalten jeweils zwei separate Transceiver-Bausteine (74HC245) das Controllersignal SYNC MASTER in einzelne, aber identische Signale für jede der zwölf Reihen auf. Im Gegensatz zum Controller schaffen diese Transceiver die erforderlichen Ströme ohne Probleme. Da nicht alle Transceiver-Ausgänge benötigt werden, ist ein Ausgang auf ein Testpad (für Tests natürlich) hinausgeführt.
Ähnlich geht es bei dem Signal TxD_MASTER von JP1 zu. Auch hier reichen die wenigen Milliampere, die IC4 liefern kann, nicht für die 96 Schrittmotor-Module aus. Also verteilen auch hier zwei Transceiver-Bausteine das Signal auf die zwölf Motorenreihen (plus eine Leitung an den ATmega des Hubs).
Etwas anders wird mit den TxD-Signalen der einzelnen Schrittmotor-Modulreihen verfahren, die den Hub erreichen. Die zwölf Leitungen werden durch Pull-Up-Widerstände auf 5 V gehalten. Sendet ein Steuermodul Daten an den Hub, zieht es die Leitung auf einen niedrigen Spannungswert und signalisiert damit eine logische 1 (wir haben es also mit einer negativen Logik zu tun). Um nur eine Eingangsleitung des ATmega auf dem Hub verwenden zu müssen, filtert eine Logikschaltung immer nur das Signal des aktuell sendenden Moduls auf die eine RX-Leitung des ATmega.
Der Controller auf dem Hub funktioniert als steuerbare Clock und liegt in der Hierarchie in einer Ebene mit den Strängen. Durch die Logikschaltung werden die einzelnen Stränge gegeneinander entprellt und sehr viele Teilnehmer können damit angebunden werden. Auf diese Weise werden nur eine Eingangs- und eine Ausgangsleitung zur Kommunikation zwischen PC und Hub benötigt.
Die Signale TxD_Clock_Gen und RxD_Clock_Gen sind für die Inbetriebnahme des Controllers gedacht, damit die Logik überbrückt werden kann und Anschlüsse für Messungen der Übertragungsignale vor und hinter der Logik gut erreichbar sind.
Das IC4 zwischen dem PC-Anschluss und der Hub-Elektronik wurde noch nicht besprochen. Es handelt sich bei dem ISO7221 um einen schnellen Zweikanal-Busisolator, der für eine sichere Potentialtrennung von Kommunikations- und Versorgungsleitungen sorgt. An JP2 lässt sich der PC aber auch ohne Isolator anschließen.

### Die Idee der Ansteuerung

PC und Hub werden mit einem USB-zu-TTL-Wandlerkabel [6] verbunden. Die Kommunikation erfolgt dann über ein eigenes UART-Protokoll mit acht Datenbits, gerader Parität und

**Tabelle 1. Anweisungen zur Bewegung und Steuerung (Beispiele).**

| | Data0 | Data1 | Data2 | Data3 | Data4 | Data5 |
|---|---|---|---|---|---|---|
| Bewegung um x Schritte | 0xA0 | Geschwindigkeit | x Schritte High-Byte | x Schritte Low-Byte | Richtung | - |
| Bewegung zur Position x | 0xA1 | Geschwindigkeit | Position x High-Byte | Position x Low-Byte | - | - |
| Bewegung aus Binärdaten | 0xA4 | Binäre Schrittanweisung 1 | 2 | 3 | 4 | Richtung |
| Stopp der Bewegung | 0xB0 | - | - | - | - | - |
| Stopp der Bewegung und Motorstrom abschalten | 0xB1 | - | - | - | - | - |
| Fortsetzen nach Stopp | 0xB2 | - | - | - | - | - |

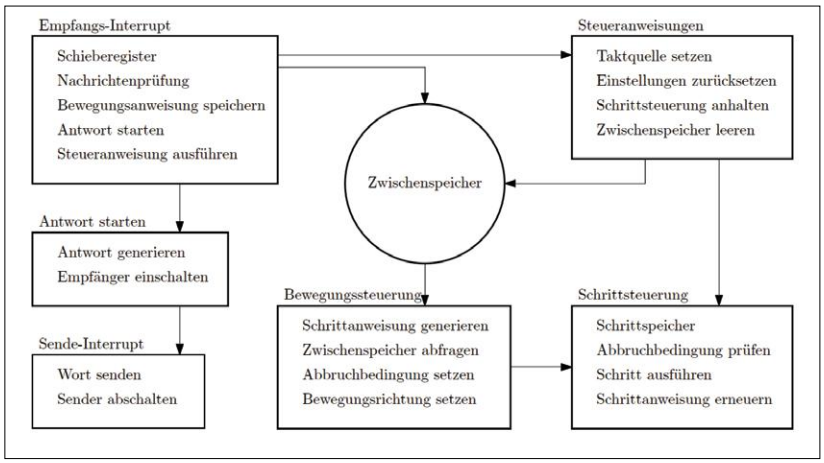



Bild 6. Abläufe der Hub-Firmware.

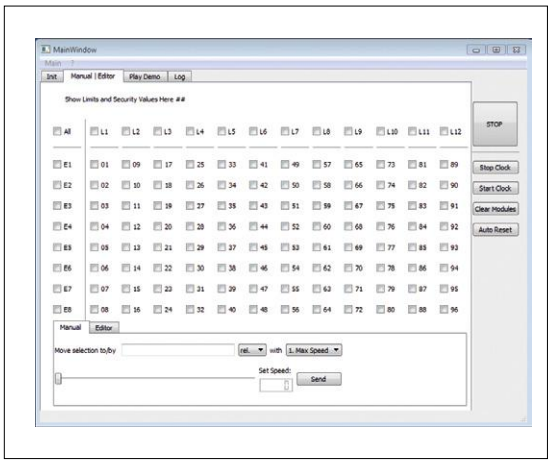


Bild 7. Komfortables GUI der PC-Software.

einem Stoppbit. Zu Testzwecken ist die Baudrate zunächst auf 600 Baud limitiert (600 8E1). Eine Nachricht an die Steuerungsplatinen auf den Motoren ist dabei wie folgt aufgebaut:

```
Adresse  Data0  Data1  Data2  Data3  Data4  Data5  ID
  CRC  STOP
```

Insgesamt können durch ein Adressbyte 254 Adressen angesprochen werden, da sowohl die Broadcastadresse 0x00 als auch 0xFF nicht für einzelne Ansteuerbefehle verwendbar sind. Jede Nachricht wird byteweise auf Paritätsfehler und einen möglichen Überlauf hin überprüft. Diese Funktionen sind bereits in der Hardware des Mikrocontrollers implementiert und können über ein Statusregister ausgelesen werden, um eventuelle Fehler zu erkennen.
Nach erfolgreicher Prüfung der Nachricht wird entschieden, ob es sich um eine Bewegungs- oder eine Steueranweisung handelt. Steueranweisungen werden direkt ausgeführt, Bewegungsanweisungen zunächst in den Speicher geschrieben. Dieses Vorgehen stellt sicher, dass alle Motoren den Steuerbefehl bereits kennen und gleichzeitig ausführen können, sobald die Software hierfür den Befehl sendet.
Nachdem der Master eine Nachricht gesendet hat, wird der Status der einzelnen Motortreiberstufen abgefragt. Sofern ein Slave nicht antwortet, wird die Nachricht erneut versendet. Der Slave antwortet seinerseits mit einer Nachricht von neun Byte Länge:

```
Adresse  Empfangene ID  Speicherplatz  Pos.H  Pos.L
  Takt H  Takt L  CRC  STOP
```

Neben der eigenen Adresse wird hier auch die zuletzt empfangene Nachrichten-ID sowie der aktuell freie Speicher, der aktuelle Takt und die Position als High- oder Low-Byte übertragen. Die Fehlererkennung erfolgt durch Berechnung eines CRC-Bytes aus allen vorangegangenen Bytes nach CCITT/ITU-T-Standard. Das Stoppbyte beendet die Nachricht.
Die bereits erwähnten Steuerungs- beziehungsweise Bewegungsanweisungen enthalten alle notwendigen Befehle, um beim Betrachter den Eindruck einer fließenden Bewegung der Skulptur zu erwecken. Einige dieser Befehle werden in **Tabelle 1** erläutert. Weitere Befehle können je nach Anwendungsfall innerhalb der Firmware unkompliziert implementiert werden. Als einfache Testroutine wurde eine Sinustabelle mit vorberechneten Anweisungen hinterlegt, die für alle Steuermodule die berechneten Werte enthält, um eine sinusförmige Bewegung der Skulptur auszuführen. Ohne vorherige Programmierung kann durch dieses Bewegungsprofil beispielsweise die Stromaufnahme, die Synchronität und die Drehrichtung der einzelnen Motoren überprüft werden.

## Die Firmware im Mikrocontroller des Hubs

Die Firmware des zentralen Mikrocontrollers ist interruptgesteuert. Ein Empfangsinterrupt bearbeitet alle eingehenden Werte und überprüft diese. Nach erfolgreicher Prüfung entscheidet die Firmware, ob eine Antwort auf die Nachricht erforderlich ist. Falls ja, wird sie generiert und vom Sender übertragen. Abschließend schaltet der Sender wieder ab.
Die Bewegungssteuerung erhält ihre Befehle aus dem Zwischenspeicher und wandelt sie in Schrittanweisungen für die Schrittsteuerung um. Zudem setzt sie Einstellungen für die Bewegungsrichtung und die Abbruchbedingungen.
Die Schrittsteuerung wird durch den Takt getriggert und sorgt für den eigentlichen Schritt, nachdem die Abbruchbedingung geprüft wurde. Die Ablaufanweisungen werden als binäre Daten im Schrittspeicher hinterlegt, sodass die Bewegungssteuerung weniger ausgelastet werden muss. Die Abläufe der Firmware sind in **Bild 6** übersichtlich dargestellt.

## Die PC-Software

Die PC-Software bietet eine komfortable Steuer- und Programmiermöglichkeit und stellt eine sinnvolle Ergänzung dar. Innerhalb der Software können Bewegungs- und Steuerungsanweisungen an den Hub gesendet sowie Bewegungsprofile eingestellt werden. Hierfür werden jeweils eine Anweisung zusammen mit der Adresse, einer Identifikationsnummer der Nachricht, einer Prüfsumme und einem Stoppbit zu einer Nachricht zusammengefasst.
Sofern nur eine Nachricht versendet wird, wird direkt im Anschluss auf eine Antwort gewartet, die wiederum sofort nach Eintreffen geprüft wird. Sofern innerhalb einer angegebenen Wartezeit keine Antwort erfolgt ist, wird die Nachricht erneut versendet, bis nach einer fest definierten Anzahl an Versuchen abgebrochen wird.
Beim Senden mehrerer Nachrichten wird für jede Nachricht ein Speicherbereich reserviert, um dort die einzelnen Antworten

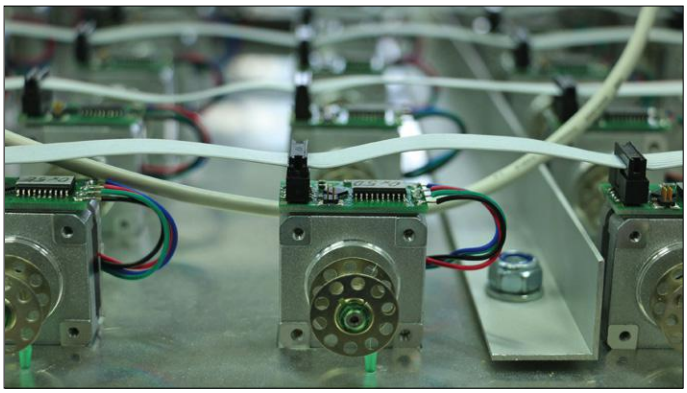
für die spätere Erfolgsauswertung zwischenspeichern zu können. Bei der Versendung von Anweisungsserien ist eine nach Adressen geordnete Sortierung der Nachrichten vorgesehen. Die Softwareoberfläche in **Bild 7** wurde für Testzwecke zunächst spartanisch gehalten, kann jedoch leicht angepasst und erweitert werden. Über ein Registermenü am oberen Rand sind die manuelle Steuerung, der Editor als Demoapplikation, ein Demo-Modus sowie ein aktueller Log verfügbar. Im Editor kann ein Bewegungsprofil für die gesamte Skulptur in die Matrix programmiert werden. Dabei werden die Höhe, in die die Kugeln fahren sollen und die Geschwindigkeit, in der dies passieren soll, eingetragen. Im Demo-Menü steht ein zu Testzwecken vorprogrammiertes Testmuster zur Verfügung, mit dem man die Funktion der Skulptur überprüfen kann. Durch die Überwachung im Log-Fenster können Fehler direkt über die Software ermittelt werden, ohne dass man mit dem Oszilloskop auf Fehlersuche gehen müsste.

### Endlich: Die Inbetriebnahme

Zu Beginn erfolgt eine Synchronisierungsfahrt, damit alle Kugeln auf einer definierten Starthöhe hängen. Alle Motoren fahren die Kugeln bis zum Anschlag herauf. Hier sorgt die Strommessung über den Shunt-Widerstand dafür, dass die Endposition durch einen erhöhten Stromfluss erkannt wird. Anschließend kann jeder Motor im Rahmen der Schrittgenauigkeit des Motors gesteuert werden.
Im Labor für Elektrische Antriebssysteme der Fachhochschule Aachen funktionierte die erste Inbetriebnahme planmäßig. Die einzelnen Motoren sind synchron zueinander ansteuerbar. Inzwischen wurden erste Tests mit einem vorgegebenen Muster erfolgreich abgeschlossen.
Die Überprüfung des gesamten Systems mit einer Wärmebildkamera zeigte keine unerwarteten Hotspots. Das elektrische Design kann daher als stabil bezeichnet werden. Inwieweit die Nylonfäden den Langzeitbetrieb überstehen, bleibt abzuwarten. Die freie Aufhängung (im Tischgestell) gewährt einen sehr leisen Betrieb.

### Fazit

Der Besuch des BMW-Museums hat bleibende Eindrücke hinterlassen und den Nachbau der Skulptur angeregt. Die FH Aachen hat dadurch ein interessantes Kunstwerk im Labor für elektrische Antriebsysteme gewonnen, das auch in Zukunft Potential für weitere Abschlussarbeiten bietet und schon heute einen



Blickfang für Studierende und Besucher darstellt.
Wer nun ein Zucken in Richtung Lötkolben verspürt, findet alles, was man zu einem Nachbau benötigt – Schaltpläne und Layouts im Eagle-Format, Software für alle Beteiligten und sogar technische Zeichnungen der Trägerplatte (natürlich nur passend für den genannten Motortyp) - auf der Elektor-Projektseite [7]. Doch auch wenn Sie sich bereits beim Nachbau für das heimische Wohnzimmer sehen, seien Sie gewarnt! Die Zeit, beinahe 100 Mal die gleiche Platine eigenhändig zu bestücken, kann man durchaus auch sinnvoll bei einer gemütlichen Runde mit Freunden verbringen. Gleichzeitig besteht bei einem solchen Bierabend weniger Gefahr, zum Opfer eines Künstlerwitzes über Elektrotechnik-Nerds zu werden, die 100 Mal die gleiche Leiterbahn aufkratzen mussten, weil doch noch ein winziger Fehler in der Platine war… ⏮

(160272)

### Weblinks


[1] https://artcom.de/project/kinetische-skulptur/
[1] https://artcom.de/en/project/kinetic-sculpture/
[2] http://bit.ly/2jimvRu
[3] http://bit.ly/2jR13nv
[4] www.lawerence.de/map/struktur.php
[4] https://en.wikipedia.org/wiki/Network_topology
[5] http://katalog.we-online.de/en/em/WP-BUTR_TWO-ROWS/7460307
[6] www.elektor.de/usb-ttl-interface-cable-5v
[7] www.elektormagazine.de/160272


## Über die Autoren

**Prof. Dr.-Ing. Michael Bragard** studierte nach seinem Abitur im Jahr 2000 Elektrotechnik an der RWTH Aachen. Nach dem Abschluss als Diplom-Ingenieur im Jahr 2006 arbeitete er als wissenschaftlicher Mitarbeiter am Institut für Stromrichtertechnik und elektrische Antriebe. Ab 2012 war er bei Delta Energy Systems im Bereich der Entwicklung von Windkraftanlagen tätig. Im Januar 2015 erfolgte der Ruf auf die Professur für das Lehrgebiet „Elektrische Antriebssysteme“ an der Fachhochschule Aachen.

**Andreas Lambers, B.Sc.** studierte nach dem Abitur 2006 Elektrotechnik an der RWTH Aachen und beendete das Studium der Elektrotechnik an der FH Aachen als Bachelor of Engineering. Das Thema der Abschlussarbeit war die „Entwicklung und Inbetriebnahme einer skalierbaren Visualisierungsplattform aus Schrittmotoren mit serieller Steuerung“.

# Fernbedienung mit Bluetooth Low Energy

## Mit nur einem Knopfdruck...

Von **Jennifer Aubinais** (F)

Im Bluetooth-Protokoll ist festgeschrieben, dass ein Master mit einem oder mehreren Slaves kommuniziert. Eine Kommunikation zwischen den Slaves ist nicht möglich. Der Master wird meistens von einem Smartphone oder Computer verkörpert, doch nicht immer ist dies die optimale Lösung. Die Rolle des Masters kann auch von einem Bluetooth-Modul übernommen werden.

Im Verlauf des Jahres 2015 hat Elektor mehrere Projekte zum Thema „Bluetooth Low Energy“ (BLE) veröffentlicht. Im Mittelpunkt der Beitragsreihe [1] stand das Bluetooth-Modul BL600 von Laird Technologies, das ausschließlich im Slave-Betrieb arbeitet. Ein Pendant zum BL600 ist das Modul BL620 [2], es kann die Rolle des Masters übernehmen und mit vier Slaves kommunizieren. In diesem Beitrag zeigen wir, wie das BL620 als Fernbedienung für den „Bluetooth-230-V-Schalter“ (Elektor Dezember 2015 [3]) eingesetzt werden kann. Das Bedienen des Bluetooth-Netzschalters ist mit der Fernbe-

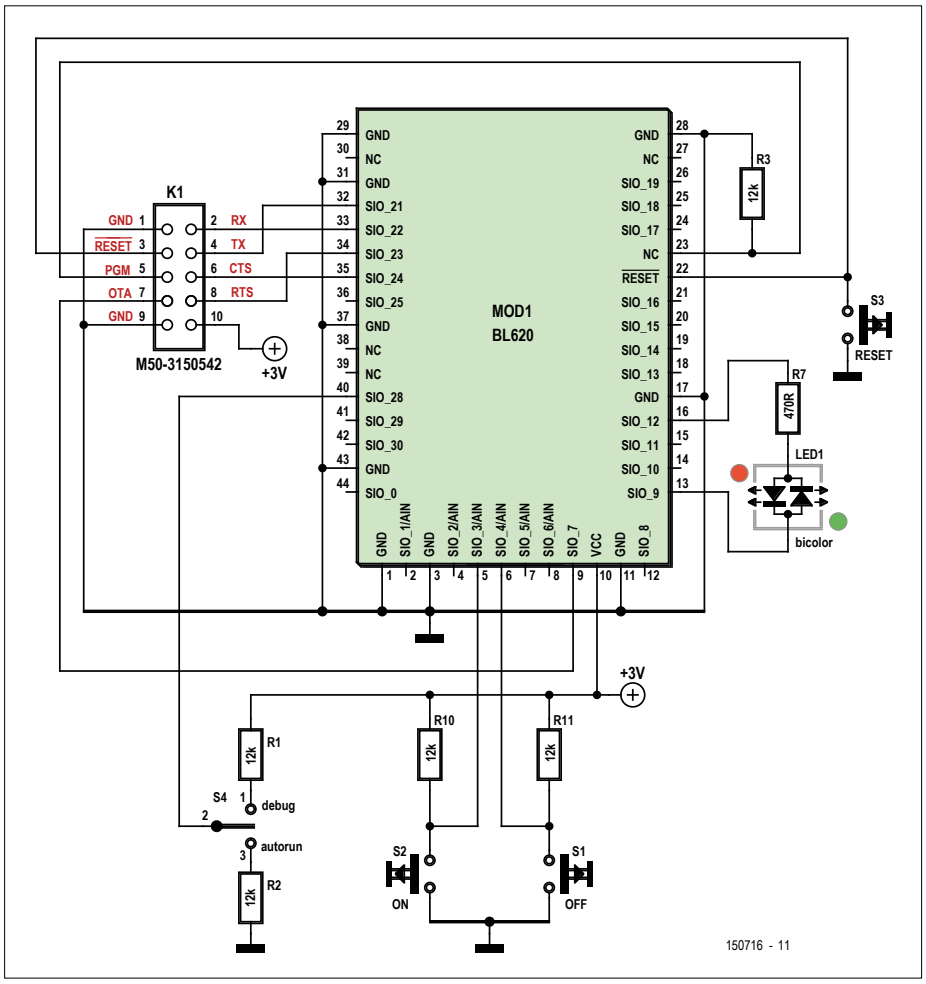



Bild 1. Die Grundschaltung der Bluetooth-Fernbedienung.

## INFOS ZUM PROJEKT

Bluetooth Low Energy
BLE
BL600

Einsteiger
➡ Fortgeschrittene
Experte

Etwa 2 Stunden

USB-Seriell-Wandler 3,3 V,
Lötwerkzeuge für SMDs,
J-Link Edu (optional)

Rund 35 €

## Eigenschaften

- Niedriger Energiebedarf
- Bluetooth Low Energy (BLE)
- Passt in das Gehäuse einer 12-V-Fernbedienung

dienung viel bequemer als mit einer App auf dem Smartphone.

### Hardware

Die Grundschaltung der Fernbedienung, dargestellt in **Bild 1**, ist überschaubar. Da im Modul BL620 fast sämtliche Funktionen vereint sind, ist der Bedarf an weiteren Bauteilen marginal. Wir haben zwei Drucktaster und eine zweifarbige LED hinzugefügt, ergänzt werden sie von einem Reset-Taster, einer Programmier-Kontaktleiste sowie einem Schalter, der vom regulären Betrieb in den Debug-Modus umschaltet. Nur die Stromversorgung weist einige Besonderheiten auf.

Wer an Fernbedienungen denkt, denkt an Batteriebetrieb, denn Fernbedienungen werden meistens aus Batterien mit Energie versorgt. Bei dieser Fernbedienung ist das nicht anders, die zugehörige Batterie hat sogar die Spannung 12 V. Das klingt zunächst befremdlich, zumal das Modul BLE620 an 3 V arbeitet. Die hohe Spannung ist das Ergebnis eines Kompromisses, denn es war ein Problem, auf dem Markt ein passendes Gehäuse zu finden. Dort mussten außer der Platine mit dem BL620 (Abmessungen 19 · 12,5 mm) noch zwei Drucktaster und eine LED untergebracht werden. Das einzige Gehäuse, das unseren Vorstellungen entsprach, war für den Betrieb mit einer 12-V-Batterie ausgelegt und für die Fernbedienung eines Garagen-

### Sleep-Modus des Bluetooth-Moduls

Wir hatten geplant, das Bluetooth-Modul BL620 unmittelbar an der Spannung 3 V zu betreiben. Bei dieser Konstellation würde sich das Modul während der meisten Zeit im Sleep-Modus befinden, der Strombedarf beträgt dann 0,5 µA. Da aus den im Text genannten Gründen eine 12-V-Batterie als Energiequelle dient, hängt die Stromaufnahme im Standby-Zustand vom Stepdown-Spannungsregler IC1 ab. Sein Strombedarf ist jedoch vergleichsweise hoch. Deshalb wird die Betriebsspannung beim Wechsel in den Standby-Zustand abgeschaltet. Vom Sleep-Modus des Bluetooth-Moduls wird kein Gebrauch gemacht. Falls in einem anderen Projekt der Sleep-Modus genutzt werden soll, ist der Aufruf folgender Funktionen nötig:

`GpioSetFunction(3,1,0x12)`:
Pin 3 (erster Parameter) wird als Eingang konfiguriert (zweiter Parameter). Der Wert 0x12 bedeutet, dass das Modul bei Low an diesem Eingang aus dem Sleep-Modus geholt wird.

`SystemStateSet(0)`:
Diese Funktion versetzt das Bluetooth-Modul in den Sleep-Modus. Danach kann die zugehörige Stromaufnahme gemessen werden.

Wenn das Bluetooth-Modul aus dem Sleep-Modus geholt wird, startet das Programm sofort, in gleicher Weise wie nach Drücken des Reset-Tasters.

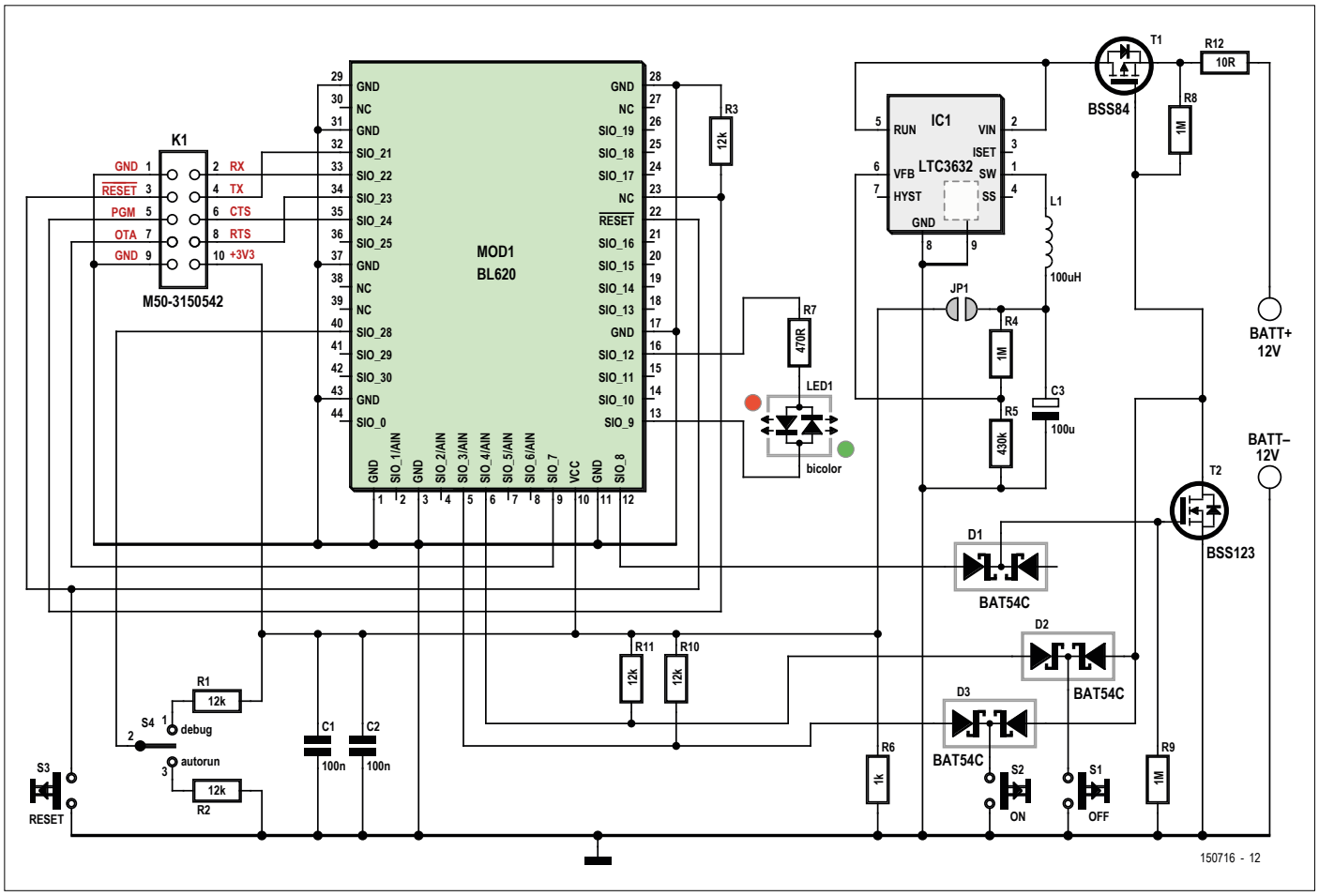



Bild 2. Vollständige Schaltung einschließlich Stromversorgung für 12 V.

tors konstruiert. Um die Spannung 12 V herabzusetzen, kommt der verbreitete Stepdown-Spannungswandler LTC3632 von Linear Technology zum Einsatz.

Damit die Batterie möglichst lange durchhält, muss der Energiebedarf der Schaltung niedrig sein. Das Modul BL620 passt hier prima, im Standby-Modus fließen nur 0,5 µA. Leider macht der Stepdown-Spannungswandler einen Strich durch die Bilanz, denn er schlägt mit 12 µA zu Buche. Das mag wenig erscheinen, doch eine 12-V-Batterie der Bauform 23A hat eine Kapazität von nur 55 mAh.

Die Fernbedienung darf deshalb nicht permanent unter Spannung stehen. Aus diesem Grund haben wir einen elektronischen Schalter hinzugefügt, der aus zwei Transistoren und drei Dioden besteht. **Bild 2** zeigt die definitive Schaltung.

Stepdown-Wandler IC1 erhält Spannung, wenn der Transistor T1 leitet. Das geschieht immer dann, wenn der Drucktaster S1 oder S2 betätigt wird. Das Bluetooth-Modul nimmt seinen Betrieb auf, indem es zuerst T2 durchsteuert. Das hat zur Folge, dass T1 im Leitzustand bleibt, solange das Modul aktiv ist, auch wenn der Taster losgelassen wird.

Die doppelten Dioden D2 und D3 haben zwei Funktionen. Zum einen schützen sie das Modul zusätzlich zur Strombegrenzung mit R8 vor der Spannung 12 V, zum anderen bilden sie eine UND-Verknüpfung, die verhindert, dass zwei Taster kurzgeschlossen werden.

Die Ausgangsspannung von IC1 wird vom Spannungsteiler R4/R5 nach folgender Beziehung bestimmt:

$V_{out} = 0{,}8 \cdot (1 + R4/R5)$

Mit den in der Schaltung angegebenen Werten liegt $V_{out}$ wegen der Toleranz der Bauteilwerte zwischen 2,8 V und 3,2 V.

Widerstand R12 ist hinzugefügt, um die Stromaufnahme messen zu können, ohne eine Leitung unterbrechen zu müssen. Wenn ein (Milli)Voltmeter an diesen Widerstand angeschlossen wird, lässt sich die Stromaufnahme der Fernbedienung unkompliziert berechnen.

Die LED1 ist eine zweifarbige LED, sie signalisiert sowohl die Bedienung des Schalters als auch den Status der Bluetooth-Verbindung (siehe **Tabelle 1**).

Außer den elektronischen Bauelementen gehören zur Fernbedienung auch

**Tabelle 1. Signale der Zweifarb-LED LED1. Vertauschte Farben bedeuten, dass die LED falsch angeschlossen ist.**

| Farbe | Blinken | Bedeutung |
|---|---|---|
| Grün | Nein | Nach Betätigen von S1 wird das Kommando „Ein“ gesendet. |
| Rot | Nein | Nach Betätigen von S2 wird das Kommando „Aus“ gesendet. |
| Grün/Rot | Langsam | Nach gleichzeitigem Betätigen von S1 und S2 wird der Bluetooth-Netzschalter gesucht. |
| Grün/Rot | Schnell | Verbindungsfehler (Bluetooth-Netzschalter nicht gefunden). |

einige elektromechanische Komponenten. Über die zehnpolige Kontaktleiste K1, verbunden mit einer seriellen Schnittstelle (3,3 V !), kann das Bluetooth-Modul BL620 mithilfe des Programms *UwTerminal* (mit einem neuen Skript) programmiert werden. Auch die (native) Firmware des Moduls ist per K1 überschreibbar. Dazu wird ein *Debug Probe J-Link* von Segger benötigt. Durch Austauschen der Firmware kann sogar ein Modul BL600 in ein Modul BL620 umfunktioniert werden (und umgekehrt). Um Platz zu sparen, hat die Kontaktleiste K1 das Rastermaß 1,27 mm. Zu diesem Projekt gehört deshalb eine kleine Adapterplatine, die dieses Raster an das Raster 2,54 mm anpasst. Die Pinbelegung der Stiftleiste entspricht der Kontaktleiste K1 des „e-BoB BL600“ [1] (**Bild 3**).

Der kleine Drucktaster S3 dient zum Rücksetzen des Bluetooth-Moduls, er kann gelegentlich nützlich sein.

Schalter S4 hat eine identische Funktion wie der Jumper JP1 des „e-BoB BL600“. In der Stellung *Autorun* nimmt das Modul beim Start den regulären Betrieb auf. Stehen Schalter oder Jumper auf *Debug* (*Cmd* beim „e-BoB BL600“), dann reagiert das Modul auf die AT-Kommandos, die über den seriellen Port empfangen werden. Diese Betriebsart ist zum Debuggen oder Experimentieren vorgesehen.

Solange Lötbrücke JP1 offen ist, wird die Stromversorgung von der übrigen Schaltung getrennt. In diesem Zustand kann vor der Inbetriebnahme die Funktion des Stepdown-Reglers IC1 überprüft werden.

## Software

Ebenso wie das Modul BL600 ist auch das Modul BL620 mit *smartBASIC* programmierbar, eine Sprache, die der Hersteller für die Module entwickelt hat. Viele der Funktionen haben wir bereits in den vorangegangenen Elektor-Beiträgen angewendet. Pflichtlektüre ist insbesondere die Beschreibung der *Events* in Teil 6 der Beitragsreihe [1]. Alle unter [1], [3], [4] und [5] gelisteten Quellen oder Weblinks sind für die Programmierung hilfreich.

Unser Programm setzt auf *smartZ* auf, eine Art Mikro-Betriebssystem, das der Hersteller bereitstellt. Wenn es in ein Modul geladen ist, können diverse Kommandos benutzt werden, die zum Anwenden des BLE-Protokolls nötig sind: hcitool lescan, hcitool lecc, gatttool --char-write… (für weitere Informationen siehe [4]). Da der Hersteller den Quellcode auf GitHub veröffentlicht hat [6], konnte er hier für die Fernbedienung angepasst werden. In **Bild 4** ist das Flussdiagramm des Programms skizziert.

Nachdem die Betriebsspannung des Moduls durch Drücken von S1 oder S2 (oder beide gleichzeitig) eingeschaltet ist, muss zuerst Pin 8 auf High gesetzt werden, damit die Betriebsspannung nach Loslassen des Tasters eingeschaltet bleibt. Anschließend gibt es zwei Möglichkeiten:

S1 und S2 werden gleichzeitig gedrückt: Das Modul sucht nach einem periphe-

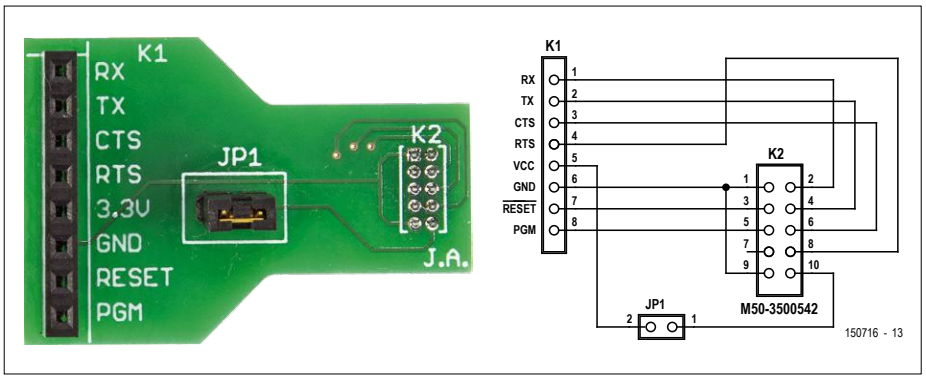



Bild 3. Das Adapter-Board passt die enge Kontaktleiste K1 an die Kontaktleiste des e-BoB BL600 an.

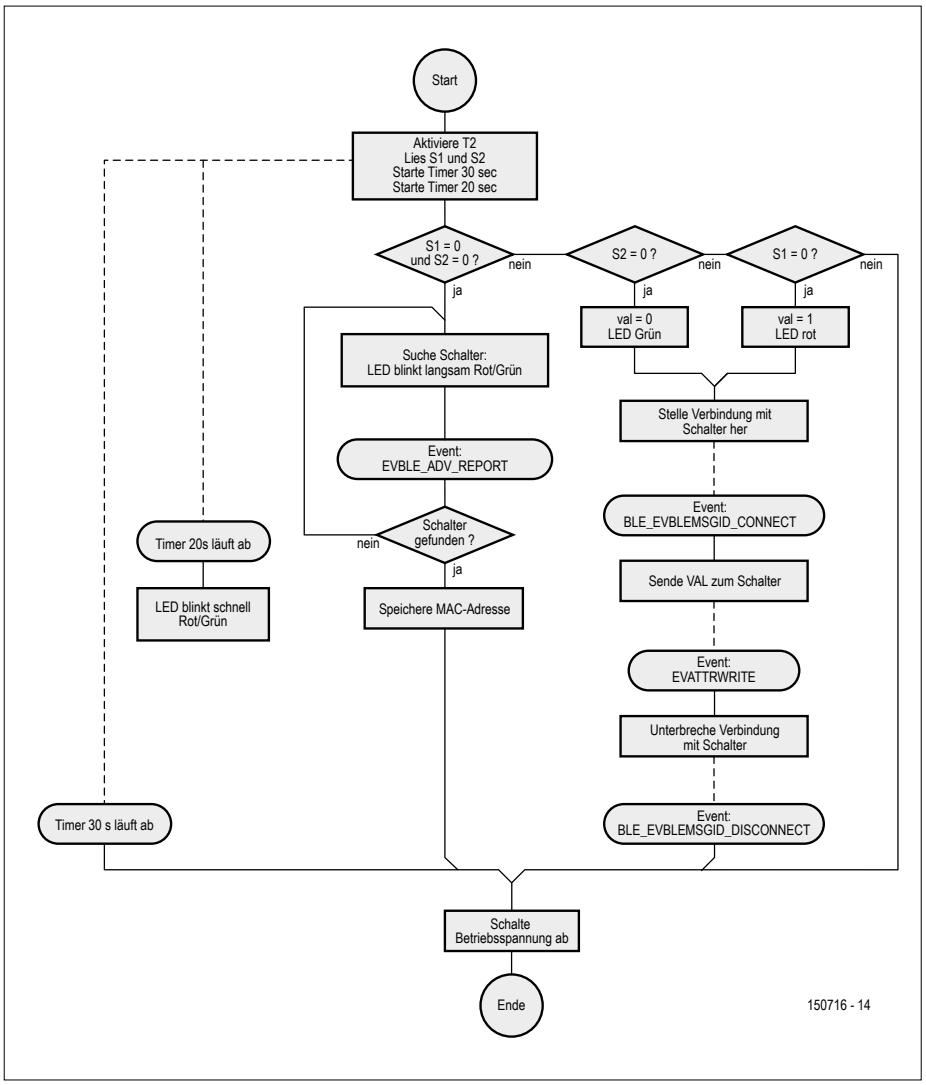



Bild 4. Flussdiagramm der Fernbedienungs-Anwendung. Der rechte Zweig stellt den regulären Ablauf dar, die durchbrochenen Linien beziehen sich auf die Events.

## STÜCKLISTE

**Widerstände:**
(SMD 0805, 5 %)
R1,R2,R10,R11 = 10 k
R3 = 12 k
R4,R8,R9 = 1 M
R5 = 365 k
R6 = 1 k
R7 = 470 Ω
R12 = 10 Ω

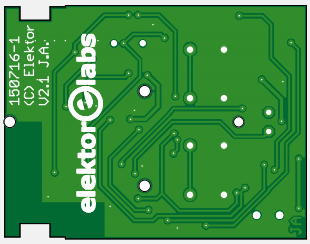
**Kondensatoren:**
C1,C2 = 100 n, 0805
C3 = 100 µ, 5650

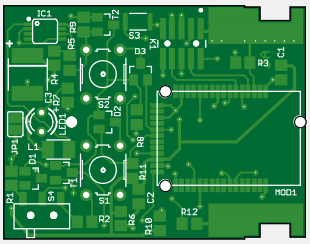
**Induktivität:**
L1 = 10 µH, 1210 (z.B. Würth Elektronik 744764910)

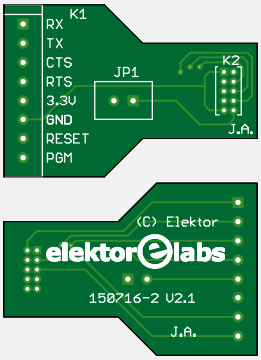
**Halbleiter:**
D1,D2,D3 = BAT54C
IC1 = LTC3632
LED1 = LED, zweifarbig, 3 mm
T1 = BSS84
T2 = BSS123

**Außerdem**
K1 = Bus-Anschlussleiste, 2 · 5 Kontakte, RM 1,27 mm, Harwin M50-3150542
MOD1 = Bluetooth-Modul BL620
S1,S2 = Drucktaster 6 · 6 mm, z.B. B3F-1020
S3 = Drucktaster Panasonic EVPAWCD2A
S4 = Schiebeschalter, C&K Components AYZ0102AGRLC
Batterie 12 V, Bauform 23A
Gehäuse CamdenBoss 2957-23 (einschließlich Batterieanschluss)

**Für das Adapter-Board**
K1 = Stiftkontaktleiste 6-polig
K2 = Stiftkontaktleiste 2 · 5-polig, RM 1,27 mm, Harwin M50-3500542
JP1 = Stiftkontaktleiste 2-polig mit Jumper








### Quellen und Weblinks


[1] „e-BoB BL600“, Teil 1 bis 6, Elektor März bis September 2015

[2] www.lairdtech.com/products/bl620

[3] „230-V-Schalter mit Bluetooth ansteuern“: www.elektormagazine.de/140115

[4] Interface von Bluetooth 2 nach Bluetooth LE: www.elektormagazine.de/150753

[5] Drahtloses Außenthermometer mit Bluetooth Low Energy: www.elektormagazine.de/140190

[6] https://github.com/LairdCP/BL620-Applications

[7] www.elektormagazine.de/150716


ren Bluetooth-Gerät, das den Namen des Bluetooth-Netzschalters (JA_SWITCH) hat. Wenn dieses Gerät gefunden ist, wird seine MAC-Adresse im EEPROM des BL620 gespeichert. Damit ist die Fernbedienung startklar, um den Netzschalter zu bedienen. Diese Aktion kann beliebig oft wiederholt werden, beispielsweise wenn der Netzschalter nicht sofort gefunden werden kann.

S1 *oder* S2 werden gedrückt: Das Modul versucht, eine Verbindung mit dem Bluetooth-Netzschalter herzustellen. Wenn die Verbindung steht, wird das Kommando gesendet, das den Schalter aktiviert (S1 gedrückt) oder deaktiviert (S2 gedrückt). Danach hebt das Modul die Verbindung auf und schaltet seine Betriebsspannung ab, indem es Pin 8 auf Low setzt. Das Aktivieren der Fernbedienung dauert weniger als eine Sekunde, weil die MAC-Adresse bereits im EEPROM des Moduls gespeichert ist und nicht mehr nach dem Netzschalter gesucht werden muss.

Wenn das Programm startet, wird ein Timer auf die Zeit 30 s gesetzt. Bei korrekter Funktion hat sich die Fernbedienung lange vor Ablauf dieser Zeit selbst abgeschaltet. Falls das nicht geschehen ist, sorgt der Timer dafür.

### Montage

Die Platine dieser Fernbedienung wird zwar mit SMDs bestückt, trotzdem ist die Montage von Hand mit einem Lötkolben und einer bleistiftfeinen Lötspitze möglich.

Die Zweifarb-LED wird etwa 1 cm über der Platinenfläche montiert, so dass sie durch eine Bohrung im Gehäuse ragt und von außen sichtbar ist. Lötbrücke JP1 auf der Platine bleibt vorläufig offen.

### Adapter-Board

Beim Adapter-Board müssen die Kontaktleiste K1 und der Jumper JP1 auf der Oberseite (Bestückungsseite) montiert werden, während die kleine Kontaktleiste K2 auf der Unterseite (Lötseite) montiert wird.

### Programmieren

Laden Sie zuerst die Software einschließlich des Programmier-Tools *UwTerminal* und der Bibliotheken herunter [7]. Setzen Sie S4 in die Stellung *Debug*, das ist die Stellung, die vom Modul BL620 weg zeigt. Schließen Sie das Adapter-Board an K1 der Fernbedienung an und verbinden Sie das Adapter-Board über ein USB-Seri-

ell-Wandler-Kabel (3,3 V !) mit Ihrem Computer. Stellen Sie im Geräte-Manager von Windows den zugehörigen COM-Port fest und starten Sie das Terminal-Programm *UwTerminal* mit 9600 Baud und der gefundenen COM-Port-Nummer.
Klicken Sie mit der rechten Maustaste und wählen Sie im Menü *Download* ➞ *BASIC* ➞ *Run* aus.
Wählen Sie die Datei *$autorun$.Remote.smartZ.sb*. Das Programm wird jetzt in das Modul geladen.
Wenn der Programmiervorgang beendet ist, schließen Sie das Terminalprogramm und entfernen Sie die Verbindungskabel. Bringen Sie S4 in die Stellung *Autorun*, in dieser Stellung zeigt der Schalter in Richtung des Moduls. Setzen Sie die Platine und die 12-V-Batterie in das Gehäuse ein und prüfen Sie, ob die Spannung am Ausgang des Stepdown-Wandlers (JP1, Pin 2, auf der Seite von IC1) innerhalb des Betriebsspannungsbereichs 1,8...3,6 V des Moduls liegt. Wenn dies zutrifft, schließen Sie die Kontakte von JP1 mit einem Tropfen Lötzinn, so dass auch die übrige Schaltung Betriebsspannung erhält.
Die Fernbedienung ist jetzt startklar. Betätigen Sie zuerst beide Taster gleichzeitig, damit sich die Fernbedienung mit dem Bluetooth-Netzschalter verbindet. Die zweifarbige LED blinkt langsam, um anschließend zu verlöschen. Wenn die LED schnell blinkt, hat die Fernbedienung den Netzschalter nicht gefunden. Im regulären Betrieb drücken Sie S1, um den Netzschalter zu aktivieren, oder S2, um ihn zu deaktivieren. ⏮

(150716)gd

## Aus Modul BL600 wird Modul BL620

Wenn ein Bluetooth-Modul BL600 verfügbar ist, aber ein Modul BL620 fehlt, können Sie das BL600 (Slave) in ein BL620 (Master) verwandeln, indem Sie es neu programmieren. Dazu brauchen Sie einen *Debug Probe J-Link* von Segger, die *Edu-Version* genügt.

- Laden Sie die BL620 Firmware v12.4.10.0 Revision 1 von www.lairdtech.com/products/bl620 ➞ Software Downloads herunter. Sie müssen sich anmelden, registrieren Sie sich mit „Elektor“ als Company.
- Laden Sie den Treiber Software for Windows von www.segger.com/downloads/jlink herunter und installieren Sie ihn.
- Verbinden Sie den *Debug Probe J-Link* wie nebenstehend angegeben mit dem BL600:

| BL600 | J-Link (Edu) |
|---|---|
| PGM | TCK |
| RES | TMS |
| GND | GND |
| VCC | VCC |

- Legen Sie das BL600 an eine externe Betriebsspannung, denn der *Debug Probe J-Link* stellt keine Betriebsspannung bereit.
- Starten Sie die Datei *_DownloadFirmwarev12_4_14_0.bat*, die zum Download aus Schritt 1 gehört.

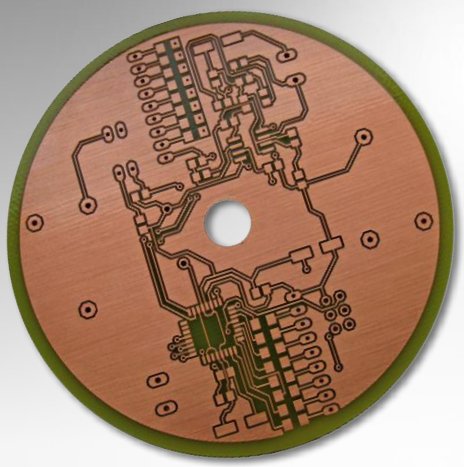
Glückwunsch, damit haben Sie Ihr Bluetooth-Modul BL600 in ein Modul BL620 verwandelt! Beschriften Sie das Modul mit „BL620“.

In umgekehrter Richtung können Sie auch ein BL620 in ein BL600 rückverwandeln. Dazu müssen Sie die neueste Firmware für das BL600 von der Hersteller-Website herunter- und in das Modul hineinladen.

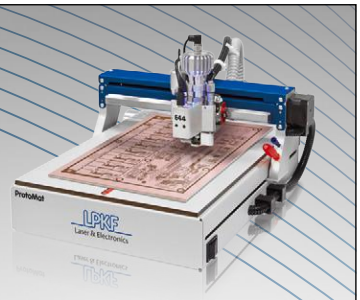
## EINKAUFSZETTEL

- ➞ **150716-1 :** Platine für Fernbedienung + Adapterplatine
- ➞ **150716-2 :** Platine für das Adapter-Board
- ➞ **140270-91 :** e-BoB BL600
- ➞ **140190-91 :** BLE-Thermometer
- ➞ **140115-71 :** BLE-Netzschalter 230 V
- ➞ **150168-91 :** Modul BL600

Anzeige



Sicher, präzise und günstig



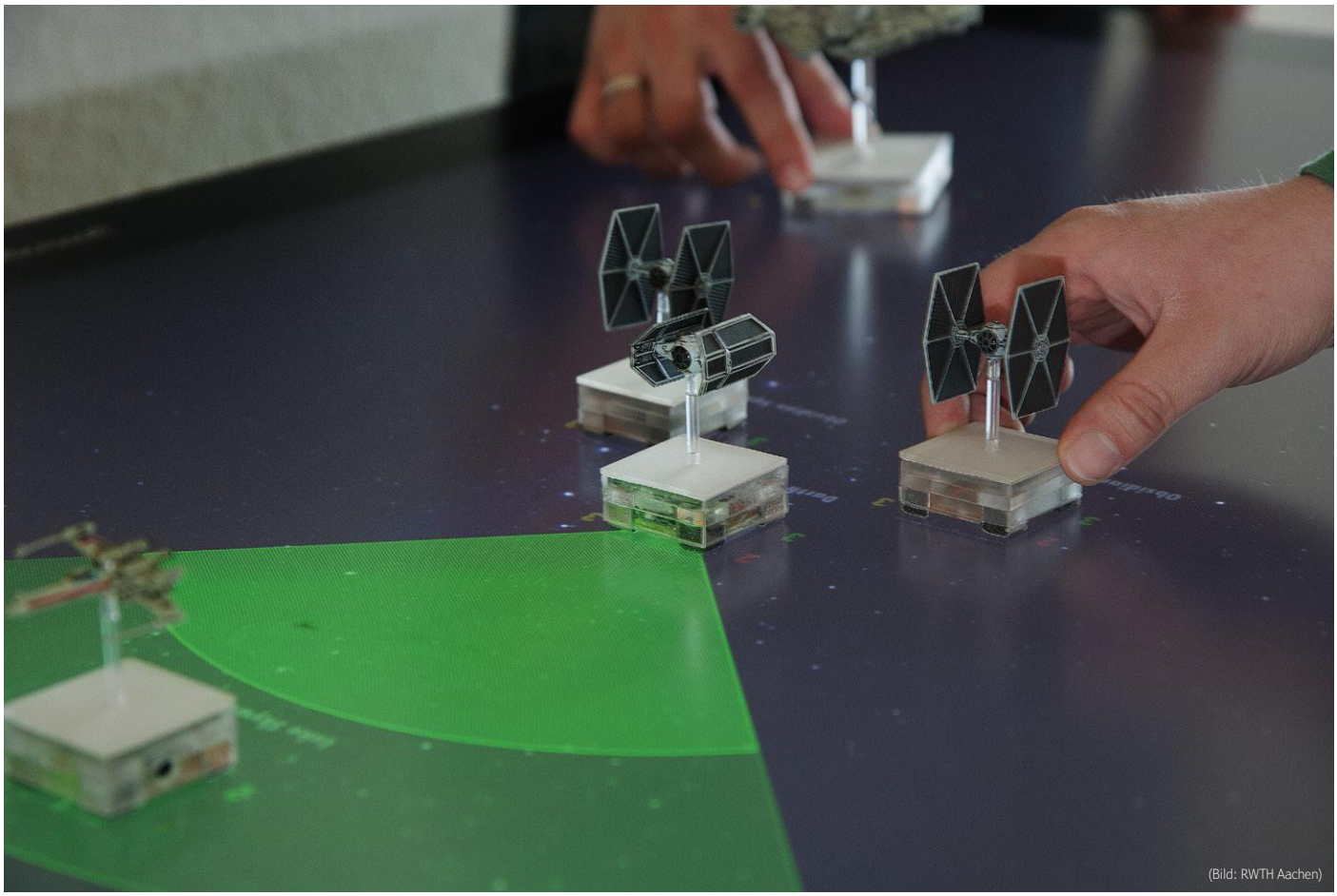
LPKF-Qualität zum Einstiegspreis: Ein- und doppelseitige Leiterplatten im eigenen Labor herstellen. Erfahren Sie mehr: **www.lpkf.de/protomat-e44**

LPKF Laser & Electronics AG Tel. +49 (0) 5131-7095-0

LPKF Laser & Electronics

# Das TABULA-Projekt

## Objekte auf Touchscreens kapazitiv erfassen


(Bild: RWTH Aachen)

Von **Christian Cherek**, Lehrstuhl für Informatik 10 (Medieninformatik und Mensch-Computer-Interaktion) an der RWTH Aachen

Große Touchscreens, auf denen anfassbare Objekte (Tangibles) abgesetzt und bewegt werden können, eignen sich unter anderem für Lernanwendungen. Bisher war es aber nicht möglich, hierfür die kapazitive Berührungserkennung zu nutzen, die in modernen Tablets und großen Touch-Displays wie dem „Surface Hub“ zum Einsatz kommt. Dies wird ein Forschungsprojekt der RWTH Aachen ändern, an dem auch Elektor beteiligt ist.

In Deutschland fehlen zurzeit mehr als 40.000 IT-Angestellte. Das betrifft nicht nur Experten, im Grunde benötigt jeder moderne Beruf IT-Kenntnisse. Deswegen muss die Informatikausbildung dahingehend verbessert werden, dass noch mehr Menschen daran teilhaben können. Das vom Bundesministerium für Bildung und Forschung (BMBF) geförderte TABULA-Projekt [1] hat sich zum Ziel gesetzt, die manchmal recht trockene Materie intuitiver und weniger abstrakt zu vermitteln.

Die dahinter stehende Idee ist, dass Lernende anfassbare Objekte auf einem (liegenden) touchfähigen Display bewegen können, das interaktiv Lerninhalte darstellt – um die Objekte herum und auch dazwischen. Hierfür müssen dem Steuerrechner der Ort und die Identität dieser „Tangibles“ genannten Objekte zu jeder

Zeit bekannt sein. Will man ein modernes Tablet oder besser noch ein großes touchfähiges Display nutzen, das Berührungen kapazitiv erkennt, dann steht man gleich vor mehreren Herausforderungen.

### Problemstellung

Die Tangibles werden von den Nutzern auf dem Display abgesetzt und bewegt. Mögliche Anwendungen reichen von der Musikproduktion über die Informatikausbildung (Suchbäume, Graphen) bis hin zu medizinischen Lehrsimulationen. Das haptische Erlebnis und das Feedback bei „Benutzereingaben" ermöglicht dabei eine neue Art der Interaktion. Es gibt bereits ähnliche Systeme, die aber meist reflektiertes Infrarotlicht nutzen, um Berührungen und die Tangibles zu erkennen. Ihre Zuverlässigkeit hängt deswegen stark von den Lichtverhältnissen ab. Direkte Sonneneinstrahlung zum Beispiel stellt ein erhebliches Problem dar.
Aus diesem Grund geht man gegenwärtig von den optischen zu kompakten und gegen Umgebungseinflüsse wenig empfindlichen kapazitiven Systemen über. Kapazitive Multi-Touchscreens (die also mehrere Berührungen gleichzeitig erkennen können) enthalten Transmitterelektroden, die in schneller Folge nacheinander aktiv geschaltet werden, und räumlich davon getrennte, im rechten Winkel zu den Sendern angeordnete Empfängerelektroden (**Bild 1**). An jeder Kreuzung bildet sich so ein elektromagnetisches Feld zwischen den Elektroden. Wenn ein Objekt mit hoher Kapazität wie ein menschlicher Finger nahekommt, ändert sich das elektrische Feld. Die Touchfläche erkennt diese Änderung und gibt eine Rückmeldung über die Koordinaten der Berührung aus, den so genannten Touchpunkt.

## Das Projekt und seine Akteure

Dieses Projekt wird vom **Bundesministerium für Bildung und Forschung BMBF** gefördert. Während der dreijährigen Projektdauer werden zwei Demonstratoren entwickelt. Der erste soll Datenstrukturen und Algorithmen aus der Informatik behandeln, während der zweite die flussbasierte Programmierung in den Mittelpunkt rückt.

Bundesministerium für Bildung und Forschung

- Die **RWTH Aachen** ist mit drei Instituten beteiligt.
  - Der Lehrstuhl für Medieninformatik und Mensch-Computer-Interaktion von Prof. Jan Borchers forscht seit vielen Jahren an modernen Interaktionsformen. Den Wissenschaftlern ist es (als einem der ersten Labore weltweit) gelungen, Tangibles für kapazitive Multi-Touchscreens zu entwickeln, die auch erkannt werden, wenn sie nicht von einem Nutzer berührt werden. Der Lehrstuhl wird zusammen mit Elektor die Tangible-Technologie weiterentwickeln.
  - Das Lehr- und Forschungsgebiet Learning Technologies von Prof. Ulrik Schroeder wird seine Erfahrungen und sein Wissen über interaktive Lernspiele und die Vermittlung von Informatikkenntnissen mit in das Projekt einbringen.
  - Und der Lehrstuhl für Kommunikationswissenschaften von Prof. Martina Ziefle wird ethische und soziale Implikationen der gewählten Technologie untersuchen.

  http://learntech.rwth-aachen.de/cms/LearnTech/Forschung/Projekte/~ltmn/TABULA/?lidx=1

- Der **Elektor-Verlag** wird zusammen mit dem Institut für Medieninformatik und Mensch-Computer-Interaktion die Tangible-Technologie verbessern und über die neuesten Designs in der Zeitschrift berichten. Außerdem sollen die Tangibles nach Abschluss des Projekts für interessierte Nutzer zum Kauf angeboten werden.
  www.elektormagazine.de/160123

- Die **Inside Unternehmensgruppe**, deren e-Author-Tool in den letzten Jahren viele Preise gewonnen hat, steuert ihr Wissen über die Erstellung von Inhalten für interaktive e-Learning-Umgebungen bei.
  www.inside-online.de/de/

Logos: gemeinfrei

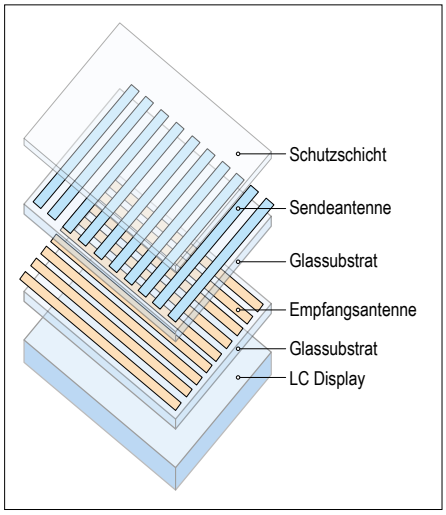



Bild 1. Aufbau eines kapazitiven Multi-Touchscreens.

Tangibles auf kapazitiven Multi-Touchscreens sind normalerweise in elektrisch leitendes Material gehüllt und/oder werden mit einem leitenden Stylus bedient, so dass die Kapazität durch den Benutzer, der sie berührt, für eine Erkennung ausreichend ansteigt. Eine leitende Verbindung von der Berührungsfläche zu den Füßen der Tangibles ermöglicht die Erkennung. Um die verschiedenen Tangibles zu unterscheiden, bilden die Füße individuelle Muster, die von der Software erkannt werden.
Auf den ersten Blick scheint dies alles wenig problembehaftet, blickt man aber auf die Details, zeigen sich die (durchaus erwünschten) Herausforderungen, denen man sich stellen muss. Ein Objekt, gerade wenn es leitfähig ist, weist nur eine sehr geringe Eigenkapazität auf, um Touchpunkte zu erzeugen. Es ergeben sich daraus zwei Nachteile:

- Tangibles auf einer kapazitiven Multi-Touch-Oberfläche werden nur dann erkannt, wenn der Benutzer sie gerade berührt. Wenn der Benutzer ein Tangible loslässt und die Kapa-

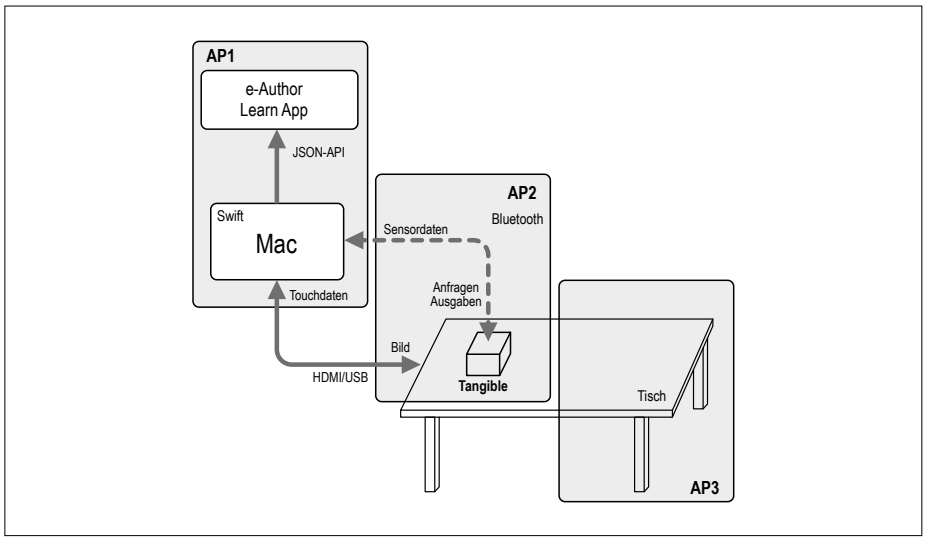



Bild 2. Das TABULA-Setup mit seinen drei Teilen Steuerrechner, Tangibles und Touchscreen.

zität abfällt, kann das System nicht unterscheiden, ob der Benutzer gerade das Tangible nur losgelassen oder aufgenommen und vom Touchscreen entfernt hat.

- Eine Erkennung der einzelnen Tangibles *ausschließlich* über ihre individuellen Touchmuster ist nur für wenige Tangibles möglich.

Beide Probleme wurden bereits im Vorfeld des TABULA-Projekts gelöst. Bevor wir in die Details gehen, können wir einen Blick auf **Bild 2** werfen, das den grundsätzlichen Aufbau des Setups (Hard- und Software) zeigt. Das Projekt besteht aus drei Teilen. Der Touchscreen (AP3) sendet seine Touchdaten über USB an einen Steuercomputer (AP1) und erhält gleichzeitig von diesem - auch von der Position der Tangibles abhängige - Bildinformationen. Über allem thront die (Lern-) Applikation, die die Rechnerdaten für den Nutzer in einer Form verfügbar macht, die der jeweiligen konkreten Anwendung entspricht. Die Tangibles (AP2) auf dem Touchtisch verhielten sich vor Beginn des TABULA-Projekt zunächst recht passiv, haben im Laufe der Projektentwicklung aber gelernt, eine aktivere Rolle zu spielen. Sie geben dem Rechner nämlich über eine Bluetooth-Verbindung Informationen preis, unter anderem über ihren Standort und ihre Identität. Schauen wir uns also an, wie sich das Projekt im Detail entwickelt hat!

## Erkennung

2013 hat der Lehrstuhl für Medieninformatik und Mensch-Computer-Interaktion an der RWTH Aachen eine Möglichkeit gefunden, das erste der obengenannten Probleme zu umgehen. Wenn man Tangibles mit leitfähigen Markerfüßen ausstattet und diese untereinander zum Beispiel mit einer Kupferfolie leitend verbindet, können die Touchpunkte auch erkannt werden, wenn ein Nutzer das Tangible loslässt.

Der Trick dahinter ist: Wenn das Messsignal nach einem Touchpunkt unter dem ersten Fuß sucht, befinden sich die anderen Füße über nicht aktiven Sendeelektroden der Touchfläche. Diese können durch die Massekapazität des Touchscreens selber eine genügend hohe Kapazität am gemessenen Punkt erzeugen. Die Massekapazität des Touchscreens ist hoch genug, dass diese durch die leitende Verbindung zwischen den Füßen (Markern) von inaktiven Sensoren zu dem gerade messenden aktiven Sensor weitergeleitet wird. Das funktioniert wie bei einem Stylus, nur dass nicht der Nutzer die Kapazität liefert, sondern der Tisch selber. Bringt man die Füße zum Beispiel in einem Drei-Punkt-Muster (rechtwinkliges Dreieck) an, dann können die Tangibles permanent erkannt werden.

Diese Lösung schafft jedoch ein neues Problem. Moderne kapazitive Systeme besitzen starke Filtermechanismen, um sich gegen Störeinflüsse aus der Umgebung zu wappnen. Diese Filter sorgen dafür, dass ein statischer Touchpunkt (wie an einem nicht berührten Tangible) nach einiger Zeit weggefiltert wird. Dadurch entsteht wieder das Problem, dass das System nicht zwischen weggefilterten und nicht vorhandenen Tangibles unterscheiden kann.

Doch zwei Jahre später, im Jahr 2015, hatte die RWTH eine Lösung in Form aktiver Tangibles gefunden. Diese neuen Tangibles wurden mit kleinen Antennen und Kantendetektoren ausgestattet, die das Messsignal der Touchfläche aufnehmen und so erkennen, wenn sie sich auf einer Touchfläche befinden. **Bild 3** zeigt ein Signal, wie man es mit dem Oszilloskop direkt auf der Oberfläche eines kapazitiven Multi-Touchscreens messen kann. Ein solches Signal wird in regelmäßigen Abständen (meist im 60-Hz-Takt) erzeugt, um Touchpunkte zu erkennen. Dieses Signal fängt das Tangible mit einer einfachen Antenne auf und filtert nur Signalstärken oberhalb der gestrichelten Linie heraus, so dass ein Signal wie unten im Bild übrig bleibt.

Die gestrichelte Linie, die Grenzwertspannung, wird mit dem Poti des Kantende-

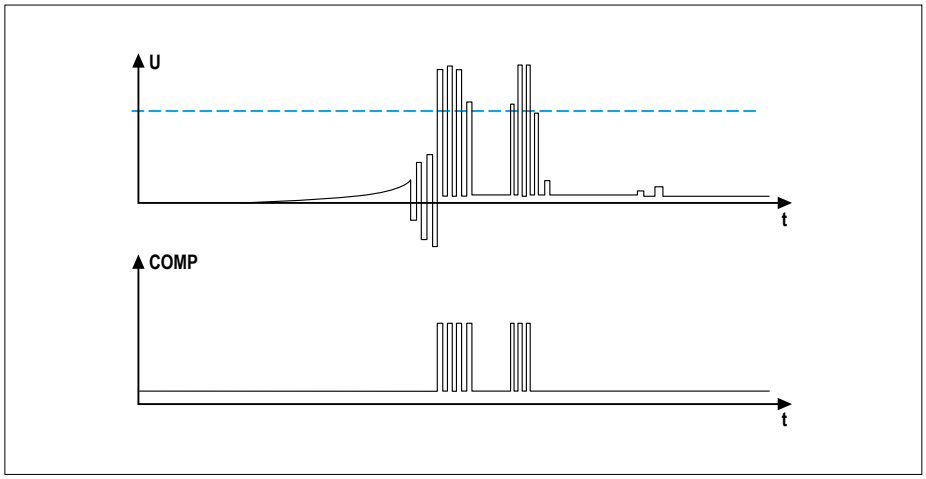



Bild 3. Oben das Signal, mit dem die kapazitive Multi-Touch-Fläche nach Toucheingaben sucht, unten das, was davon nach der Kantendetektion am Eingang des Mikrocontrollers übrigbleibt.

tektors eingestellt, dessen Schaltung in **Bild 4** zu sehen ist. Die Einstellung ist erforderlich, da die Signalstärke der Multi-Touchfläche je nach Modell verschieden sein kann.
Zusätzlich sind die Tangibles mit einer lichtempfindlichen Diode als Fallback-Lösung ausgestattet (**Bild 5**), um Lichtsignale auf dem Screen erkennen zu können. Auch hier kann die Empfindlichkeit beziehungsweise eine Schwell-Helligkeit an einem Trimmpoti eingestellt werden.
All diese Informationen sendet das Tangible per Bluetooth an den Steuerrechner, der so zwischen einem ausgefilterten und einem entfernten Tangible unterscheiden kann. Angenehmer Nebeneffekt ist, dass der Bluetooth-Chip dem Tangible eine individuelle UUID verleiht, so dass mehrere Tangibles (auch mit dem gleichen Markermuster) leicht voneinander unterschieden werden können. Damit wäre auch das zweite der obengenannten Probleme gelöst!

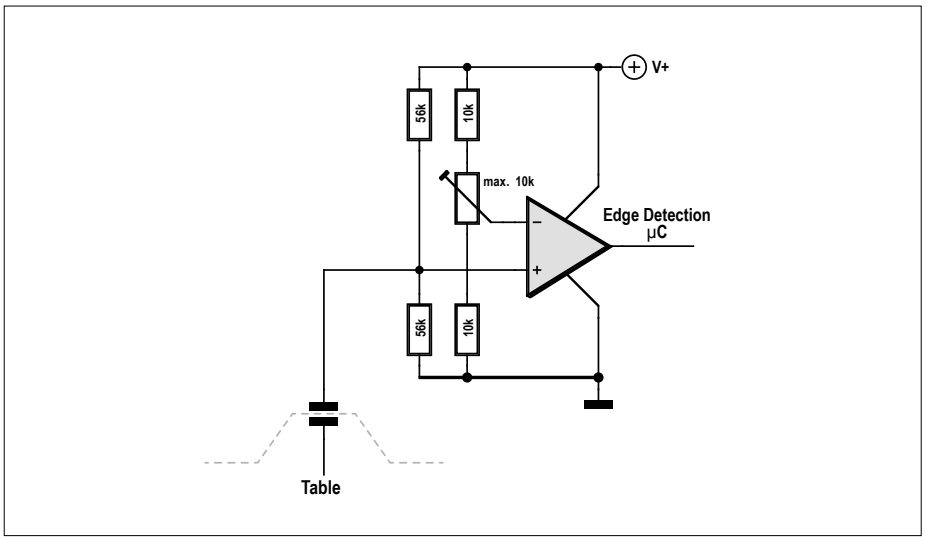



Bild 4. Schaltung für die Signalerkennung am Tangible. Am Poti wird die Grenzwertspannung eingestellt.

## Software

Softwareseitig besteht das System aus zwei Teilen, der Software auf dem mit dem Multi-Touch-Tisch verbundenen Computer und der Firmware der Tangible-Mikrocontroller. Die Software auf dem Computer nutzt die direkten Daten aus dem Touchsensor des Tisches, um potentielle Positionen der Tangibles zu erkennen. Sieht der Tisch einen Touchpunkt, sucht die Software nach vorher trainierten Touchmustern. Die Muster erzeugen die Tangibles durch die miteinander verbundenen Marker-Füße.
In unserem Aufbau ist der Steuer-PC ein iMac, auf dem der Treiber für den Touchscreen die Rohdaten der Touchauswertung an ein Swift-Framework (Multitouch Kit) weiterleitet, das dann seinerseits Touchpunkte und Tangibles voneinander unterscheidet. Da die Tangibles ein festes Muster von Touchpunkten erzeugen, kann das Framework die Position und auch die Ausrichtung eines Tangibles aus den Touchpunkten ableiten.
Die Software auf den Tangibles wird verwendet, um Verwechslungen mit Fingern oder anderen Tangibles, die dasselbe Muster erzeugen, zu vermeiden. Um die Software möglichst einfach und vielseitig einsetzen zu können, wurde als Controller die aktuelle Version eines ATmega85 verwendet. Als Bluetooth-Modul ist ein HM-11 BLE verbaut, um bei der Kommunikation mit dem PC möglichst wenig Energie zu verbrauchen. So kommt ein Tangible mit einem Energiespeicher von drei vollgeladenen AAA-Zellen auf eine ungefähre Laufzeit von 24 Stunden.

## Kantendetektor

Wenn ein Tangible in die unmittelbare Nähe des Touchscreens kommt, erkennt dessen Software das Signal des Tisches (Kantenerkennung) und schickt ein `ON-table`-Event an den Steuer-PC. Mit einem Timer überprüft das Tangible, ob regelmäßig ein Signal vom Tisch gemessen wird. **Bild 6** zeigt die Routine für diese Messung. Immer, wenn die Edge-Detection ein Signal meldet, wird der Timer zurückgesetzt.
Das Multitouch-Kit kann so die Touchpunkte einem eindeutig identifizierbaren Tangible zuordnen. Sollte das Tangible eine längere Zeit am selben Ort bleiben und die Touchpunkte vom Touchscreen nicht mehr erkannt werden, weiß die Software im Steuer-PC anhand der gesendeten Sensordaten dennoch, dass das Tangible immer noch auf dem Tisch steht. Erst wenn das Tangible vom Tisch genommen wird und nach dem Ablauf des 40-ms-Timers kein Signal vom Tisch erfolgt, schickt die Software des Mikrocontrollers ein `OFF-table`-Event, sodass der Steuer-PC das Tangible aus seiner Anwesenheitsliste streichen kann. Da der Displaytreiber Berührungen nicht nur als reinen Datenpunkt, sondern auch als Spur für bewegte Touches weitergibt, können die zu einem Tangible gehörenden Punkte auch später zugeordnet werden. Ein sich bewegendes Tangible ist in der Regel sogar leichter zu verfolgen, da bewegte Touchpunkte normalerweise sehr gut erkannt werden. Alle Berührungen, die das Multitouch-Kit keinem Tangible zuordnen kann, werden als menschliche Finger interpretiert, sodass auch weiterhin manuelle Multi-Touch-Eingaben möglich sind.
Alle diese Messungen und Vergleiche werden mit derselben Geschwindigkeit wie die Toucherkennung durchgeführt, so dass die Nutzer von der Kommunikation zwischen den Systemen kaum etwas bemerken dürften.

## Lichtsensor

Wenn ein Nutzer zufällig mit den Händen das gleiche Muster wie ein Tangible erzeugt oder zwei Tangibles gleichzeitig auf dem Tisch abgestellt werden, weiß die Software auf dem Steuer-PC nicht, welche Touches zu welchem Bluetooth-Signal gehören. In solchen Fällen

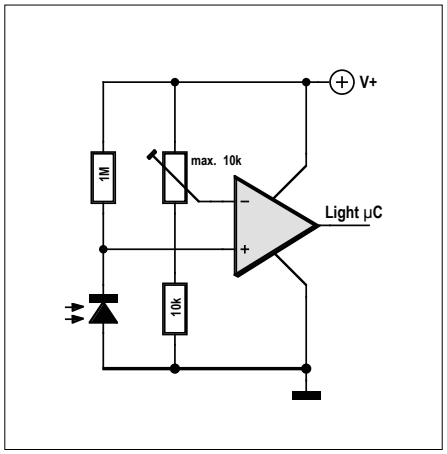



Bild 5. Schaltung für den Lichtsensor des Tangibles. Auch hier wird die Grenzwertspannung am Poti eingestellt.

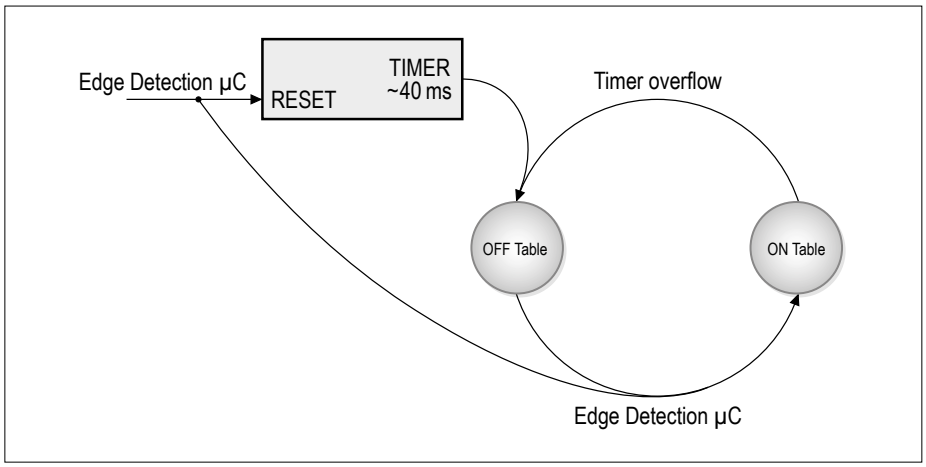



Bild 6. In der State-Machine für den Kantendetektor setzt jedes erfasste Signal den 40-ms-Timer zurück und sendet ein ON-table-Event.

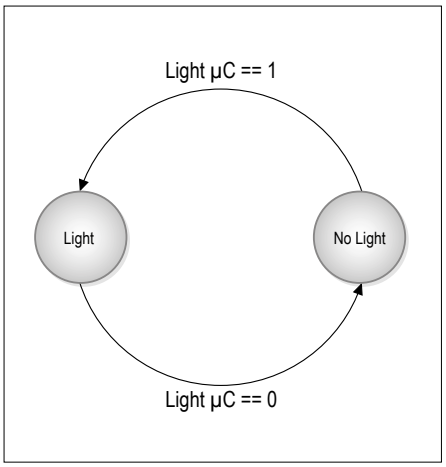



Bild 7. Die State-Machine für den Lichtsensor. Nur ein sehr heller Punkt auf dem Screen erzeugt ein Light-Event.

**Listing 1. Baumstruktur des Protokolls, mit dem die Tangibles ihre Statusinformationen an den Tisch senden.**

```
controller tree protocol
 controller
 |
 +- [0: // advanced communication port
 |
 +- [1: // product identification
 | +- 0: // manufacturer (like: rwth_i10)
 | +- 1: // product id (like: 0A4F)
 | +- 2: // uid (like: 0FA457EF)
 |
 +- [2: // basic attributes
 | +- 0: // on table?
 | | +- 0] // off table
 | | +- 1] // on table (with side 1)
 | | +- 2] // on table (with side 2)
 | | +- .. // and so on..
 | |
 | +- 1: // see light? (the side that is on the table)
 | +- 0] // see no light
 | +- 1] // see light
 |
 +- [3: // orientation, position
 | +- 0:FFFFFFFFF] // orientation (single rotation vector xxxyyyzzz)
 | +- 1:FFF] // orientation relative to table
 (| +- 2:FFFFFFFFF] // position)
 |

Examples:
 [2:0:1] // placed on the touch table
 [2:0:0] // taken off the touch table
 [2:1:1] // sees the light
 [2:1:0] // sees no light
```

und auch, wenn der Tisch bei mehreren Tangibles die Positionen überprüfen will, erzeugt die Steuersoftware auf dem Screen kleine, sehr helle Lichtpunkte an den potentiellen Tangible-Positionen. Sobald ein Lichtpunkt unter einem Tangible auftaucht, erkennt dieses den Lichtpunkt und sendet ein `LIGHT`-Signal an den PC. So kann das System entscheiden, ob und welche Touchpunkte zu einem Tangible gehören.

Da der Lichtsensor nur für besonders deutliche Helligkeitsunterschiede zuverlässige Daten liefert, sendet das Tangible nur dann ein `LIGHT`-Signal, wenn das Display sehr hell eingestellt ist. Es ist nur eine binäre Unterscheidung zwischen „Licht“ und „kein Licht“ möglich (**Bild 7**) – für eine genauere Messung ist der Sensor nicht zuverlässig genug. Doch das genügt vollkommen für den angepeilten Zweck.

## Protokoll

Das Protokoll, mit dem das Tangible seine Nachrichten an den Multi-Touchscreen sendet, ist so konzipiert, dass es möglichst einfach um zusätzliche Funktionen erweitert werden kann. Die Basissoftware auf dem Tangible, wie sie hier beschrieben wird und auch im Download-Paket enthalten ist, sendet nur die Events für die beiden Sensoren an das System. Dabei entspricht die Nachricht, die das Tangible sendet, jeweils einem Zweig aus der Baumstruktur in **Listing 1**. So wird zum Beispiel die Nachricht 2:0:1 gesendet, wenn sich das Tangible auf dem Tisch befindet. Bislang senden Tangibles nur den Status `ON-` und `OFF-table` und `LIGHT` und `NOLIGHT`. Durch die Baum-

struktur kann dieses Protokoll jedoch einfach erweitert werden. So können zum Beispiel für ein würfelförmiges Tangible mehrere verschiedene `ON-table`-Events unterschieden werden, indem man 2:0:2, 2:0:3 und so weiter sendet – je nach der Seite, auf der das Objekt auf den Tisch gelegt wurde.

Die verwendeten Displays *Microsoft Surface Hub 55"* und *Microsoft Surface Hub 84"* können jeweils bis zu 100 Touchpunkte gleichzeitig erkennen. Dies schafft Raum für eine große Menge von Tangibles und Fingereingaben. Das Multitouch-Kit läuft auch unter iOS, so dass die Tangibles auch auf iPhones oder iPads verwendet werden können.

Auf beiden Systemen kann mittels Objective C oder Swift eine Anwendung entwickelt werden, zum Beispiel mit Apples API für grafische Anwendungen (SceneKit oder SpriteKit). Um einem Tangible ein digitales Objekt zuzuordnen, bindet man ein grafisches Objekt an ein Tangible, das die Bewegungen nachvollzieht und sich so automatisch mit dem Tangible bewegt. Das hat den Vorteil, dass man einfach die Engines von Apple nutzen kann, um Effekte und anderes Geschehen (mit den Updates, die iOS 11 mit sich bringt, sogar Künstliche Intelligenz) in die Anwendung einzubauen, ohne dass man dazu eine eigene Software neu entwickeln müsste. Die Anbindung findet direkt in Apples Programmierumgebung statt und man muss nur native Anwendungen schreiben, die dann die Tangibles unterstützen. Grundsätzlich funktionieren die Tangibles jedoch auch mit anderen Programmiersprachen, wenn diese Funktionen zur Verarbeitung von (Multi-)Touch-Ereignissen mitbringen. Auch in HTML 5 kann Anwendungen schnell beigebracht werden, ein virtuelles Objekt an ein Tangible zu binden.

## Was gibt es schon?

Im Sommer 2017 ist das erste Jahr des TABULA-Projekts abgelaufen – zwei Jahre liegen noch vor den Projektpartnern. Das Elektor-Labor unterstützt die RWTH bei der Entwicklung der Hardware und Firmware für die Tangibles.

Den aktuellen, kompletten Quellcode für den Mikrocontroller kann man von der Projektseite zu diesem Artikel [2] herunterladen. Der wichtige Teil der Mikrocontroller-Software besteht aus den zwei Automaten und der Implementierung des Protokolls, mit dem das Tangible die Sensordaten an den Steuer-PC sendet.

Kurz bevor dieser Artikel fertig wurde, sind auch das Platinenlayout und die Schaltung für die ersten einsatzfähigen Prototypen fertig geworden. **Bild 8** zeigt die erste „serienreife" Tangible-Elektronik. Der vollständige Schaltplan und das Platinenlayout samt Stückliste finden sich ebenfalls im Download-Paket auf der Projektseite [2]. ◀

(160123)

## Weblinks


[1] https://hci.rwth-aachen.de/TABULA

[2] www.elektormagazine.de/160123

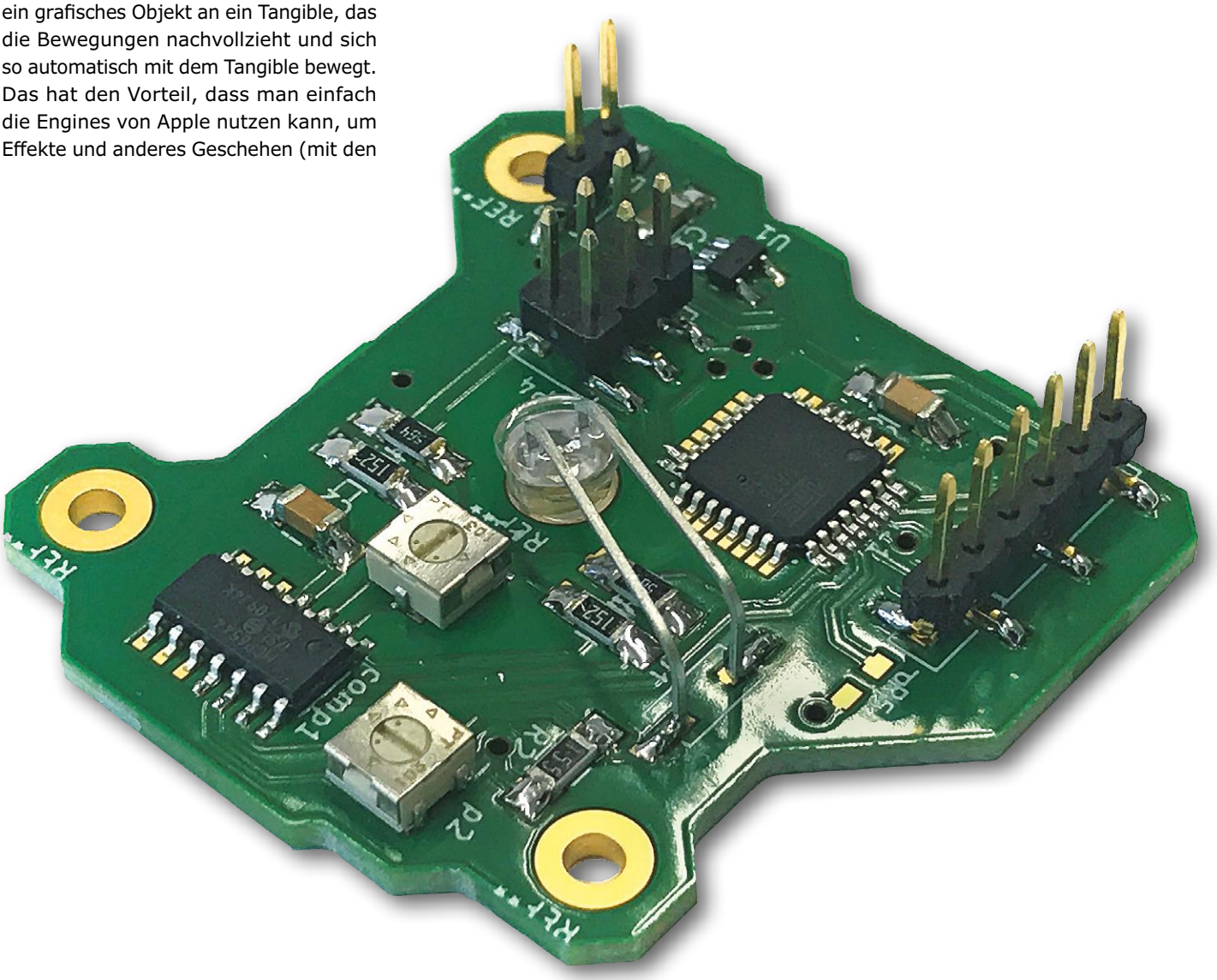
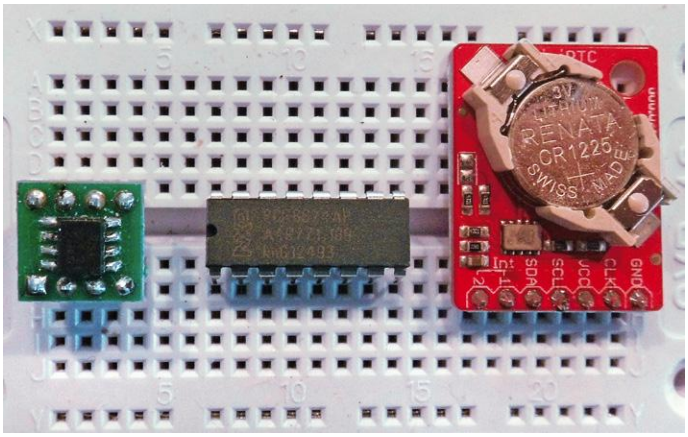




Bild 8. Die bestückte Platine mit Opamps, Trimmpotis und Mikrocontroller, aber noch ohne Bluetooth-Modul. Der Lichtsensor in der Mitte schaut nach unten auf den Touchscreen (Bild: RWTH Aachen).

# Der I²C-Bus

## Teil 3: Bausteine und Fehleranalyse



Der letzte Teil dieser Serie beschäftigt sich mit drei exemplarisch gewählten I²C-ICs: einem Temperatursensor, einem Portexpander und einer Echtzeituhr. Wir erfahren, wie ihre Register gelesen und geschrieben werden und welcher Hardware- und Software-Tools man sich dabei bedienen kann.

Von **Josef Möllers**

Es gibt viele I²C-Bausteine und -Module, vom Temperatursensor über Real-Time-Clocks und Lage- und Bewegungssensoren bis hin zu LED- und Grafikdisplays. Man braucht nur im Internet nach einer entsprechenden Liste zu suchen oder im Suchfeld der Webseite seines Elektronikhändlers einmal „I2C“ eingeben, schon erhält man seitenweise Bausteine und Module [1], von denen hier einige vorgestellt werden sollen.

I²C-Bausteine erfordern keine großartige Verkabelung mit dem Host. Mit einem vierpoligen Flachbandkabel für Daten und Spannungsversorgung sind sie schnell angeschlossen. Im Titelbild dieses Artikels sind die drei hier behandelten I²C-Bausteine nebeneinander auf einem Steckbrett zu sehen, von links nach rechts: ein Temperatursensor LM75 auf einer Adapterplatine, ein I/O-Expander PCF8574 und ein RTC-Modul mit dem Echtzeituhr-Chip RV-8523 links unterhalb der Batterie.

### LM75

Der LM75 ist **der** Standard-Temperatursensor mit I²C-Bus-Ansteuerung. Von seiner 7-Bit-Adresse sind nur die oberen vier Bits (1001) fest vorgegeben, die drei unteren Bits werden durch externe Beschaltung festgelegt. An einem I²C-Bus lassen sich also acht LM75 mit den 7-Bit-Adressen 0x48 bis 0x4F betreiben. Setzt man den LM75 zum Beispiel zur Temperaturkontrolle in einem großen Rechner ein, könnte man so bis zu acht dieser Sensoren im Gehäuse verteilen.

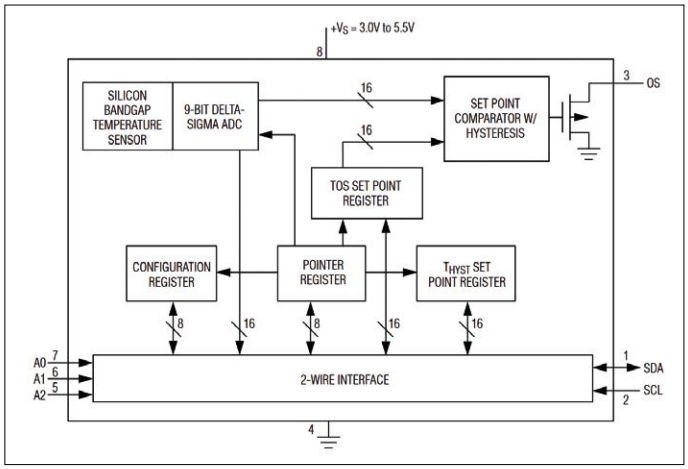



Bild 1. Innenleben des Temperatursensors LM75 (Bild: Maxim).

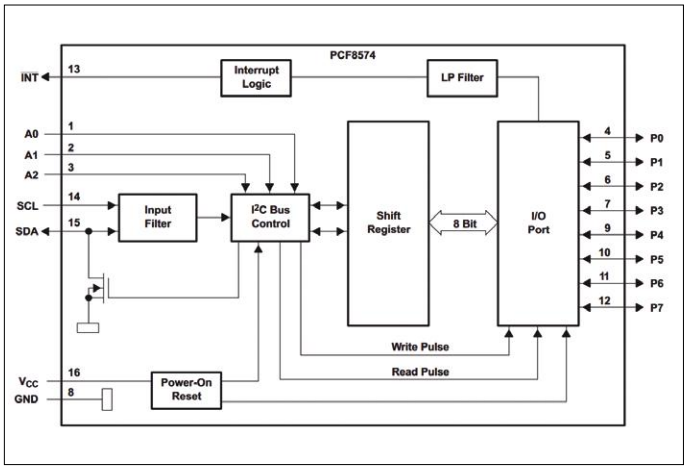



Bild 2. Innenschaltung des Portexpanders PCF8574 (Bild: Texas Instruments).

Intern verfügt der LM75 über vier Register, die über zwei Bit adressiert werden (**Bild 1**):

- 00H: ein 16 Bit breites Temperatur-Register, das nur gelesen werden kann,
- 01H: ein 8 Bit breites Konfigurations-Register,
- 02H: ein 16 Bit breites Hystereseregister,
- 03H: und ein 16 Bit breites Vergleichs-Register.

Obwohl nach dem Einschalten das Temperatur-Register angewählt ist, sollte man immer, wenn man ein Register (auch das Temperaturregister) lesen möchte, zuerst die Registernummer übertragen. Beim Schreiben muss die Registernummer immer übertragen werden, denn das erste an einen LM75 übertragene Byte wird immer als Registernummer interpretiert. Danach folgen die Daten: Bei den 16 Bit breiten Registern wird das höherwertige Byte zuerst und dann das niederwertige übertragen.
Im Gegensatz zu einigen anderen I²C-Bausteinen erhöht der LM75 die Registernummer nach einem Zugriff nicht, sie bleibt auf dem gesetzten Wert. Wenn nur ein Master vorhanden ist und lediglich die Temperatur gelesen werden soll, muss zum Lesen also nicht ständig die Registernummer wieder auf 00 gesetzt werden. Es reicht dann aus, in Lese-Transaktionen nur die beiden Temperaturbytes zu übertragen.
Die Genauigkeit des LM75 lässt leider zu wünschen übrig, es gibt aber genauere Ersatztypen, die meist protokoll- und registerkompatibel zum LM75 sind, z.B. den TMP275.
Der LM75 ist übrigens sehr empfindlich, was Störungen auf der Stromversorgung angeht. Die Faustregel „Ein 100-nF-Kondensator pro Gehäuse“ sollte bei diesem Baustein also unbedingt beherzigt werden, ansonsten kann es passieren, dass der Master Datenmüll aufsammelt.

## PCF8574

Der PCF8574 ist ein „Remote 8-bit I/O expander“, ein paralleler I/O-Baustein mit I²C-Bus-Ansteuerung (**Bild 2**). Ihn gibt es in zwei Varianten, die sich nur in der I²C-Adresse unterscheiden: Beim PCF8574 lauten die oberen vier Bit der Adresse 0100, beim PCF8574A 0111. So können bis zu 16 dieser Bausteine an einem I²C-Bus betrieben werden.
Intern gibt es lediglich ein Register, das fest mit dem Zustand der Ports verbunden ist. Schreibt man ein Bit-Muster in das Register, ändern sich die Port-Bits. Setzt man ein Port-Bit auf „1“, kann man es auch als Input verwenden. Liest man dann das Register, so erhält man Auskunft über den logischen Zustand des externen Pins. In **Bild 3** ist ein PCF8574 mit einer LED an P0 an einen Raspberry Pi angeschlossen. Schreibt man eine „1“ in den PCF8574,

```
i2cset -y 1 0x40 0x01
```

so leuchtet die LED auf, schreibt man eine „0“,

```
i2cset -y 1 0x40 0x00
```

so verlischt sie wieder. Streng genommen wird hier eine Registernummer `0x01` beziehungsweise `0x00` übertragen, da `i2cset` laut Handbuch und Online-Hilfe nach der I²C-Adresse eine Registernummer erwartet – der Portexpander interpretiert die Werte allerdings als Bitmuster für seinen Ausgang.
Es kann manchmal sinnvoll sein, den Rechner von der Peripherie

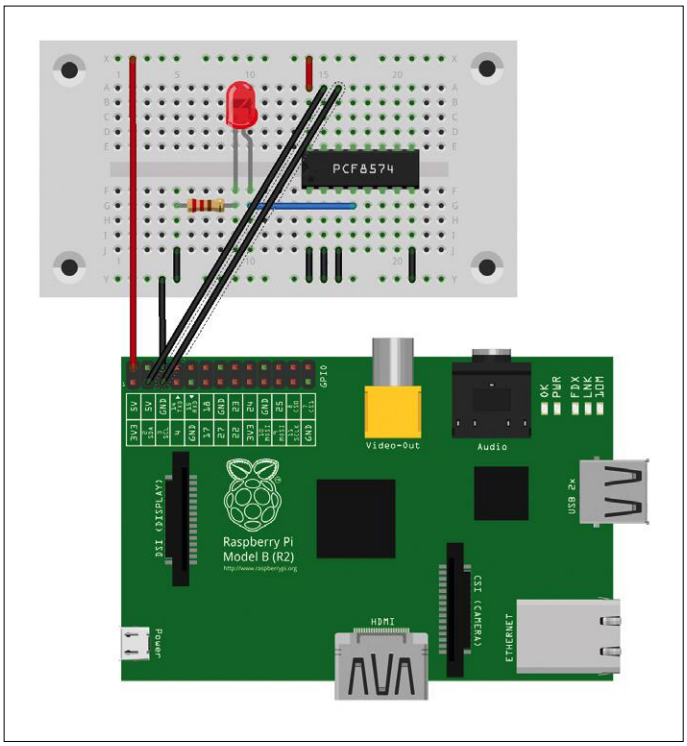



Bild 3. Der Portexpander PCF8574, angeschlossen am Raspberry Pi.

elektrisch etwas zu trennen, damit ein versehentlicher Kurzschluss zum Beispiel nach 12 V „nur“ den PCF8574 zerstört.
PCF8574 werden auch auf einfachen LCD-Interface-Platinen eingesetzt, mit denen die gängigen ein- oder zweireihigen LCDs mit HD44780-Controllern I²C-fähig gemacht werden. Das LCD wird dann im 4-Bit-Modus angesteuert, drei weitere Pins des PCF8574 steuern E, RS und R/W. Das letzte Bit kann zur Steuerung der Hintergrundbeleuchtung des LCDs eingesetzt werden. Auf manchen dieser Boards sind Pull-up-Widerstände nach 5 V vorgesehen. Diese sollte man am besten erst gar nicht bestücken beziehungsweise sofort entfernen.

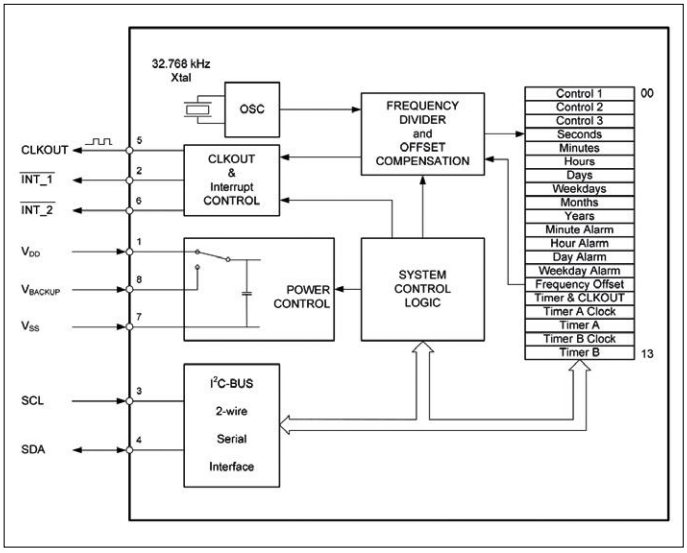



Bild 4. Die Echtzeituhr verfügt über 14 adressierbare Steuer-, Zeit- und Alarmregister (Bild: Micro Crystal).

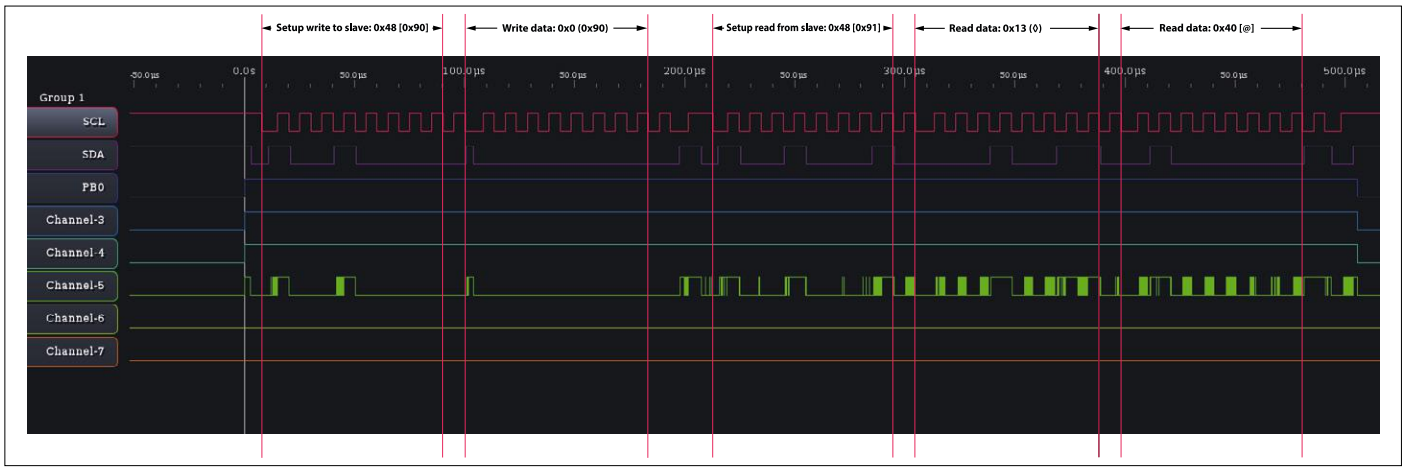



Bild 5. Der Open Logic Sniffer beobachtet, wie die Temperaturwerte eines LM75 von einem ATmega88 gelesen werden.

## RV-8523

Die RV-8523 ist eine Echtzeituhr (RTC, Real Time Clock) mit 20 Registern á acht Bit mit den Nummern 00H bis 13H, die rechts in **Bild 4** aufgeführt sind. Die RTC verfügt über eine interne Spannungsüberwachung und kann selbsttätig die Spannungsversorgung auf eine Batterie umschalten. Wie im Titelbild zu sehen, wird die RTC oft auf einem Modul komplett mit Batteriehalter geliefert.

Nachdem man eine Registernummer (zwischen `0x00` und `0x13`) übertragen hat, kann man auf das Register zugreifen. Im Gegensatz zum LM75 schreibt der Baustein die Registernummer fort, mit einem Überlauf von `0x13` zurück nach `0x00`. Man kann also mit einem Zugriff die gesamten 20 Register auslesen oder beschreiben.

Geht es aber zum Beispiel nur um Datum und Uhrzeit, setzt man die Registernummer auf 3 und liest sieben Bytes, wie in diesem Arduino-Wire-Beispiel:

```
Wire.beginTransmission(0x68);
 Wire.write(byte(0x03));    // set register number to 3
 Wire.endTransmission();
 Wire.requestFrom(0x68, 7); // read time and date
 seconds = Wire.read();
 tenseconds = (seconds >> 4) & 0x07; seconds &= 0x0f;
 minutes = Wire.read();
 tenminutes = (minutes >> 4) & 0x07; minutes &= 0x0f
 ...
```

Die Werte sind schon BCD-kodiert, eine Umwandlung von binär nach dezimal entfällt also.

Die RV-8523 besitzt neben der eigentlichen Uhr auch eine Weckfunktion, die zu einem vorgegebenen Zeitpunkt einen Interrupt produziert. Das Problem ist nur, dass der INT_1-Ausgang zwar auf LOW geht, wenn der Zeitpunkt gekommen ist, aber nicht aus eigenen Stücken wieder zurück auf HIGH. Die Alarm-Bedingung muss explizit zurückgesetzt werden (AF im Control-Register 2).

Einige Betriebssysteme wie Raspbian bringen einen Treiber für diesen Baustein schon mit (`rtc_pcf8523`), so dass eine eigene Programmierung gar nicht nötig ist, so lange es nur um die aktuelle Uhrzeit geht: Das Kommando `hwclock` greift auf die RTC zu und liest/setzt die Uhrzeit. Ein `rc`-Skript beim Systemstart führt das Kommando aus und setzt die Systemuhr, beim Herunterfahren wird die (eventuell per Internet oder manuell) aktualisierte Systemzeit wieder in die RTC geschrieben. In dem Fall kann man einem Raspberry Pi, der ohne weitere Netz-Anbindung läuft, mit minimalem Hardware-Aufwand die Uhrzeit über einen Stromausfall hinweg retten. Will man die Weckfunktion benutzen, kommt man aber um eine eigene Programmierung nicht herum.

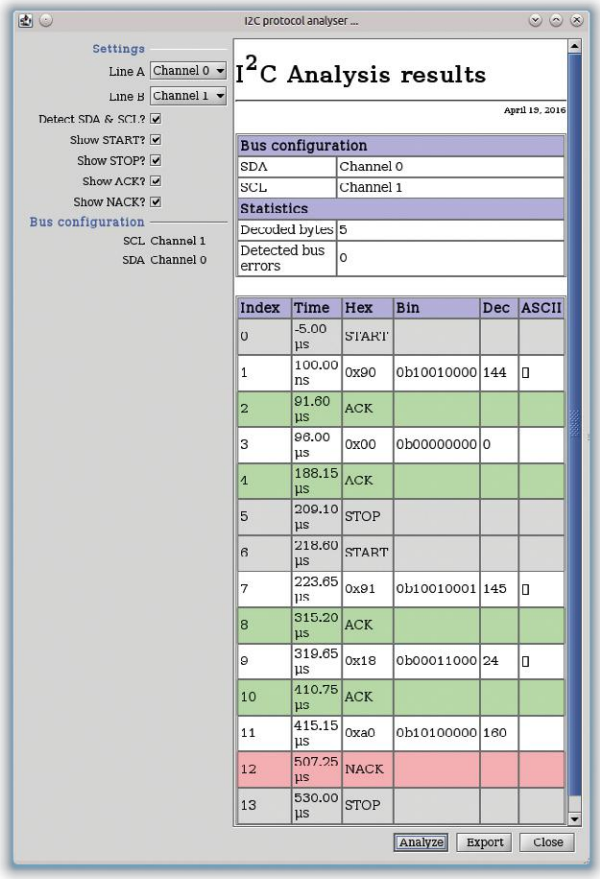



Bild 6. Der OLS Java Client stellt die Busparameter in Tabellenform dar.

Wenn man die Uhr mit dem Raspberry Pi setzt, kann man sie danach an einen ATmega oder Arduino anschließen. Die Pufferbatterie auf dem Modul sorgt dafür, dass dabei die Uhrzeit weiterläuft. So lassen sich mit wenigen Zeilen Code Datum und Uhrzeit für einen ATmega oder Arduino nutzen.

### Fehlersuche, Fehleranalyse

Nicht immer funktioniert alles auf Anhieb. Mit einem Multimeter kann man die Fehlersuche beginnen, aber wenn es um Protokoll-Probleme geht, reicht das meist nicht.
Im Ruhezustand liegen SDA und SCL auf HIGH. Zuerst stellt man also mit einem normalen **Multimeter** fest, ob SCL und SDA nach dem Einschalten (und am besten im Ruhezustand des Busses) einen genügend hohen Pegel aufweisen. Wie bereits oben erwähnt, spezifiziert der Standard zwar einen HIGH-Pegel von mindestens 0,7 $V_{cc}$ (bei 5 V also mindestens 3,5 V), viele 5-V-Bausteine funktionieren jedoch auch mit Pull-Up-Widerständen nach 3,3 V, da die HIGH-Pegel in der 3,3-V-Logik in der Regel doch deutlich über 3 V liegen. Vielleicht lässt sich das Problem ja schon mit ein oder zwei Widerständen lösen.
Stimmen die Spannungen, muss man zu anderen Mitteln greifen. Es gibt verschiedene **Logikanalysatoren** für wenig bis (sehr) viel Geld. Ob es sich nun um einen kleinen ScanaQuad [2] oder einen ausgewachsenen Red Pitaya [3] handelt oder, wie bei mir, einem Open Logic Sniffer [4] von Dangerous Prototypes mit dem OLS-Java-Client von ols.lxtreme.nl [5]: Mit allen kann man dem Datenverkehr auf dem I²C-Bus auf den Grund gehen.
**Bild 5** zeigt, wie ein ATmega88 das Temperaturregister eines LM75 liest. Zu Anfang (bei 0.0 s) sieht man die *Start-Condition*, am Ende (bei 500.0 us) die *Stop-Condition* und bei 200.0 µs die *Repeated-Start-Condition*. Zwischen dem Triggerpunkt und etwa 200.0 µs wird der LM75 zum Schreiben adressiert und die Registernummer `0x00` an den LM75 übertragen, nach der *Repeated-Start-Condition* wird der LM75 nochmal, diesmal zum Lesen, adressiert und es werden die beiden Bytes `0x13` und `0x40` aus dem Temperatur-Register an den Atmega88 übertragen. Der Trigger erfolgt übrigens über PB0: Bevor die Übertragung beginnt, wurde PB0 auf 1 gesetzt, nach der Übertragung wieder auf 0.
Der OLS-Java-Client analysiert die Aufzeichnung und die I²C-Kommunikation und zeigt die übertragenen Bytes in tabellarischer Form an (**Bild 6**). Fälschlicherweise wird die Repeated-Start-Condition als Stop-Condition, gefolgt von einer Start-Condition angezeigt, obwohl im Diagramm deutlich zu sehen ist, dass bei 200.0 µs keine Stop-Condition gesendet wird.
Der **Bus Pirate** von Dangerous Prototypes [6] in **Bild 7** ist ein Werkzeug zum Analysieren von seriellen Protokollen wie I²C, SPI oder einem UART-Datenverkehr. Dangerous Prototypes verkauft den Bus Pirate und den Open Logic Sniffer zwar selber, stellt die Hard- und Software aber auch frei zur Verfügung, so dass man beide Module sogar selbst bauen kann. Die Version 4 des Bus Pirates ist laut Hersteller „auf die Zukunft ausgerichtet“, läuft aber nicht so stabil wie die Version 3.6, die beim deutschen Distributor Watterott Electronic [7] erhältlich ist.
**Tabelle 1** zeigt die Adressierung und die Datenübertragung. Man sieht, dass der LM75 seine Adresse in beiden Fällen (Schreiben und Lesen) und die Registernummer `0x00` mit ACK quittiert. Dies gilt auch für die Register-Nummer, denn es könnten ja noch mehr Bytes übertragen werden, auch wenn das Temperatur-Register nur lesbar ist. Der ATmega88 quittiert dann das

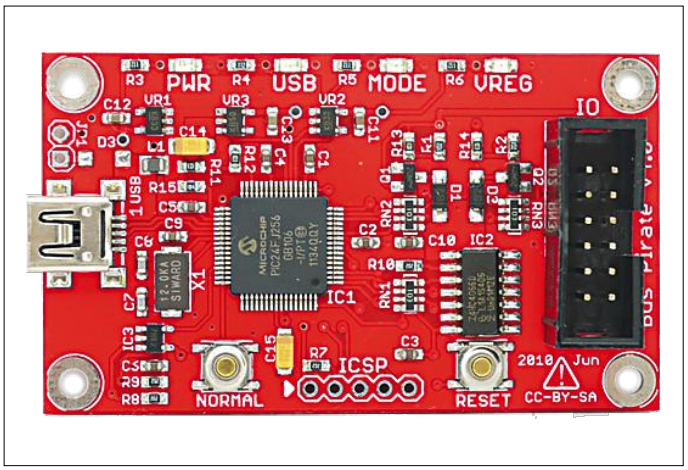



Bild 7. Der kleine Bus Pirate ist ein universelles Businterface für den PC (hier die neue Version 4; Quelle: DangerousPrototypes).

erste der beiden Bytes aus dem Temperatur-Register mit ACK, weil er noch mehr erwartet und das zweite Byte mit NACK, weil es das letzte ist, was übertragen werden soll. Danach beendet der ATmega88 die Kommunikation. Der Bus Pirate kann nicht erkennen, wer ACK oder NACK gesendet hat.
Der Bus Pirate kann auch die Pull-up-Widerstände (10 kΩ) bereitstellen. Bei der Version 3 muss dazu Pin 5 (VPU) des

**Tabelle 1. Adressierung und Datenübertragung bei einem LM75.**

| | |
|---|---|
| `HiZ>`**`m`**<br>`1. HiZ`<br>`2. 1-WIRE`<br>`3. UART`<br>`4. I2C`<br>`5. SPI`<br>`6. 2WIRE`<br>`7. 3WIRE`<br>`8. LCD`<br>`9. DIO`<br>`x. exit(without change)` | Rufe das Menü auf … |
| `(1)>`**`4`**<br>`Set speed:`<br>`1. ~5KHz`<br>`2. ~50KHz`<br>`3. ~100KHz`<br>`4. ~400KHz` | … I²C Bus … |
| `(1)>`**`3`**<br>`I2C READY` | … 100 kHz Geschwindigkeit |
| `I2C>`**`(2)`**<br>`Sniffer`<br>`Any key to exit`<br>`[0x90+0x00+][0x91+0x13+0x40-]` | Starte den Bus-Sniffer |

*fett: Eingabe*
[ *deutet eine* START condition *an,*
*die Zahlen geben die übertragenen Werte wieder,*
*+ ein ACK,*
*- ein NACK und*
] *eine* STOP condition.

| Tabelle 2. Lesen eines LM75-Sensors mit dem Bus Pirate als Master. | |
|---|---|
| I2C>**[0x90**<br>I2C START BIT<br>WRITE: 0x90 ACK | Sende *START condition*, Adresse 0x48 und Schreib-Bit (0), der LM75 antwortet mit ACK |
| I2C>**0x00**<br>WRITE: 0x00 ACK | Sende Byte 0x00, der LM75 antwortet mit ACK |
| I2C>**]**<br>I2C STOP BIT | Sende *STOP condition*. |
| I2C>**[0x91 r:2**<br>I2C START BIT<br>WRITE: 0x91 ACK<br>READ: 0x14 ACK 0x20 | Sende erneut *START condition*, dann Adresse 0x48 und Lese-Bit (1), der LM75 quittiert dies mit ACK. Lies danach 2 Bytes. Der Bus-Pirate quittiert das erste empfangene Byte mit ACK, das zweite jedoch erst, wenn feststeht, dass es auch das letzte Byte ist, also wenn er angewiesen wurde, die *STOP condition* zu senden. |
| I2C>**]**<br>NACK<br>I2C STOP BIT | Sende erneut *STOP condition*, da das letzte empfangene Byte auch das letzte zu übertragende war, quittiere es aber vorher mit NACK. |
| Man kann das Ganze auch in einer Zeile eingeben: | |
| I2C>**[0x90 0x00][0x91 r:2]**<br>I2C START BIT<br>WRITE: 0x90 ACK<br>WRITE: 0x00 ACK<br>I2C STOP BIT<br>I2C START BIT<br>WRITE: 0x91 ACK<br>READ: 0x14 ACK 0xA0<br>NACK<br>I2C STOP BIT | |

I/O-Headers mit der gewünschten Spannung verbunden werden. Dann liegt diese Spannung auch an einem Analogschalter, an dem alle Pull-up-Widerstände auf dem Bus Pirate angeschlossen sind. Dies ist eigentlich überflüssig, ja, bei 5 V sogar gefährlich, wenn ein Raspberry Pi zusätzlich angeschlossen ist. Bei der neuen v4-Version erfolgt die Verbindung übrigens per Software. Ob ein Pull-up-Widerstand angeschlossen ist und an welche Spannung, lässt sich leicht an den Busanschlüssen Pin 7 (SCL) und Pin 8 (SDA) des Verbinders nachmessen.
Die Pull-up-Widerstände lassen sich über ein Kommando ein- und wieder ausschalten:

```
I2C>P
Pull-up resistors ON
```
(Schalte Pull-up-Widerstände ein)

und

```
I2C>p
Pull-up resistors OFF
```
(Schalte Pull-up-Widerstände aus)

Danach kann man mit dem Bus Pirate, ähnlich wie mit dem `i2cdetect`-Kommando von Raspbian, den I²C-Bus nach Slaves durchsuchen, wobei hier die Schreib- und Lese-Funktionen getrennt angezeigt werden:

```
I2C>(1)
Searching 7bit I2C address space.
Found devices at:
0x90(0x48 W) 0x91(0x48 R)
```

Das wäre die Alternative, wenn man gerade keinen Raspberry Pi zur Hand hat, sich aber nicht sicher ist, welche I²C-Adresse ein Slave besitzt. Man kann den Bus Pirate auch als Master verwenden und das I²C-Protokoll manuell durchlaufen. In **Tabelle 2** wird beispielhaft gezeigt, wie man die Temperatur aus einem LM74 liest.

## Zusammenfassung

Der I²C-Bus ist eine extrem einfache Art, Peripherie an einen Prozessor anzuschließen, die nur ein geringes Datenvolumen und keine Interrupts erfordert. Zwei Drähte, mehr braucht es nicht! Und da es ein Multi-Drop-Bus ist, lassen sich gleich mehrere Slaves über die beiden Drähte ansprechen. Viel länger als einen Meter sollte die Verbindung aber nicht sein. Dank frei verfügbarer Bibliotheken wie der von Peter Fleury hält sich auch der Programmieraufwand in Grenzen und viele Fehlerquellen sind damit im Voraus bereits ausgeschlossen.
Die für eine Fehlersuche nützlichen Messgeräte sind entweder in den meisten Fällen vorhanden (Multimeter) oder kosten nicht die Welt. Eine Unmenge an Sensoren und anderen I²C-Bausteinen wartet geradezu darauf, Kontakt mit dem Mikrocontroller(board) auf Ihrem Labortisch aufzunehmen!

(160373)

## Weblinks


[1] zum Beispiel: http://rn-wissen.de/wiki/index.php/I2C_Chip-%C3%9Cbersicht

[2] www.elektor.de/logic-analyzer-scanaquad-sq100

[3] www.elektor.de/stemlab-125-10-starter-kit

[4] http://dangerousprototypes.com/docs/Open_Bench_Logic_Sniffer

[5] http://ols.lxtreme.nl/

[6] http://dangerousprototypes.com/docs/Bus_Pirate

[7] www.watterott.com/en/Bus-Pirate



**Danke!**

Ich danke meinen (ehemaligen) Hardware-Kollegen Franz Otte und Michael Kleineberg für Tipps und Hinweise bezüglich der Hardware sowie meinen (ehemaligen) Kollegen Reinhard Bernhardi-Grisson, Norbert Bandzius und Thomas Schlüssler fürs Probelesen und Feedback.

# Elektor Labs Pipeline

Klassiker auf elektormagazine.de/labs – aus Vergangenheit, Gegenwart und Zukunft

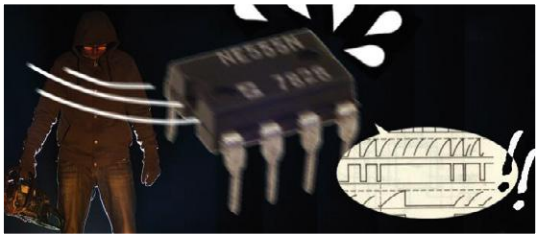
**RC5-Lautstärkesteller für den RPi-Audio-DAC**
Wer die Musik-Lautstärke mit analoger Hardware einstellen, aber nicht in teure servo-gesteuerte Potentiometer investieren möchte, findet hier einen Kompromiss in Form einer sogenannten „Digital-Controlled Analog Volume Control“. Die Schaltung verfügt über eine serielle Steuerschnittstelle, eine einstellbare Nulldurchgangserkennung und eine Stummschaltung. Diese Schaltung wurde für den Einsatz am Raspberry Pi mit Volumio entwickelt, kann aber an praktisch jeder Audioquelle verwendet werden. Es ist sogar eine Steuerung per Fernbedienung möglich.

https://goo.gl/rT3Wq0

**Bauen Sie Ihren eigenen AVR-Playground**
Vor dem Arduino hatten die Mikrocontroller-Entwicklungsboards Peripherie wie Drucktasten und LEDs, ein Display, ein oder mehrere Potentiometer für analoge Signale, Erweiterungsstecker und natürlich eine anständige Stromversorgung. Damit war eine einfache Möglichkeit zum Kennenlernen des Mikrocontrollers geschaffen, ohne dass Bauteile gelötet oder hinzugefügt werden mussten. Der AVR-Playground - Open Source, Open Hardware - wurde in diesem Sinne entworfen und erweitert den Arduino um all die kleinen Dinge, die wir zu schätzen gelernt haben.

https://goo.gl/XBGg0L

**Das Ende des 555?**
Ich liebe Zeit! Je mehr ich davon habe, desto besser! Leider ist unsere Zeit begrenzt, aber wenn man Aufgaben gleichzeitig ausführen kann, fühlt es sich an, als ob wir mehr davon haben könnten. Timer sind in solchen parallelen Prozessen wichtig, da sie Start-, Pause- und Stoppsignale ohne menschliches Zutun liefern. Der 555 ist seit Äonen der Standard-Timer aller Elektronik-Ingenieure, doch so vielseitig er auch sein mag, er ist nicht für alle Anwendungen geeignet. Dieses Projekt will das ändern.



https://goo.gl/GHwWQd

**Eine 50.-Geburtstags-Schaltung für den µA723**
Ein klassischer Chip aus ferner Vergangenheit: der Spannungsregler µA723. Laut Wikipedia wurde dieser Chip vom legendären Bob „every-idiot-can-count-to-one“ Widlar entworfen und von Fairchild Semiconductor im Jahr 1967 vertrieben. Nun, 50 Jahre später, begegnen wir dieser Schönheit in ihrer Metal-Can-Version in einer digital gesteuerten Stromquelle. Dies zeigt, dass man mit hervorragender Spannungs- und Lastregelung ein langes Leben im Scheinwerferlicht verbringen kann. ⏮

(160479)

https://goo.gl/EwGjtN

# Marderscheuche

## Mit unhörbaren 90 dB

Von **Jan Visser** (Elektor-Labor)

Der Marder ist eigentlich ein possierliches Tierchen. Menschen greift er nicht an, aber er kann ihnen gewaltig zur Last fallen: Unangenehme Gerüche, lautstarke Ruhestörungen und Vandalismus unter den Hauben fahrbarer Untersätze gehören zu seinem Repertoire. Durchgenagte Kabel der Motorelektrik sind keine Seltenheit. Weil Marder vom Gesetz geschützt sind, dürfen sie weder gefangen noch getötet werden. Was also tun?

### Eigenschaften

- Betriebsspannung: 9...12 VDC
- Mittlere Stromaufnahme: 7 mA
- Stromaufnahme bei Schallabgabe: 80 mA
- Stromaufnahme in Ruhe: 2 mA
- Frequenz (einstellbar) 20...43 kHz
- Intermittierendes Schallsignal

Ein Leser fragte beim Elektor-Labor an, ob es möglich sei, einen elektronischen Stacheldraht gegen die Marderplage zu konstruieren, unter der er permanent litt. Nun ist bekannt, dass Marder schrille Geräusche verabscheuen, wenn die Frequenzen im Bereich des Ultraschalls liegen, etwa bei 23 kHz. Für den Menschen sind solche Töne nicht hörbar. Diese Erkenntnis war der Ansatz für das Konzept unserer neuen elektronischen Marderscheuche. Ein langjähriger Kollege erinnerte sich, dass Elektor vor mehr als einem Jahrzehnt schon einmal eine elektronische Marder-Abwehr veröffentlicht hat, es war im Dezember 2002. Damals war die Schaltung noch vollständig diskret aufgebaut, auf einer vergleichsweise umfänglichen Platine. Das passt nicht in die heutige Zeit der Mikrocontroller, so

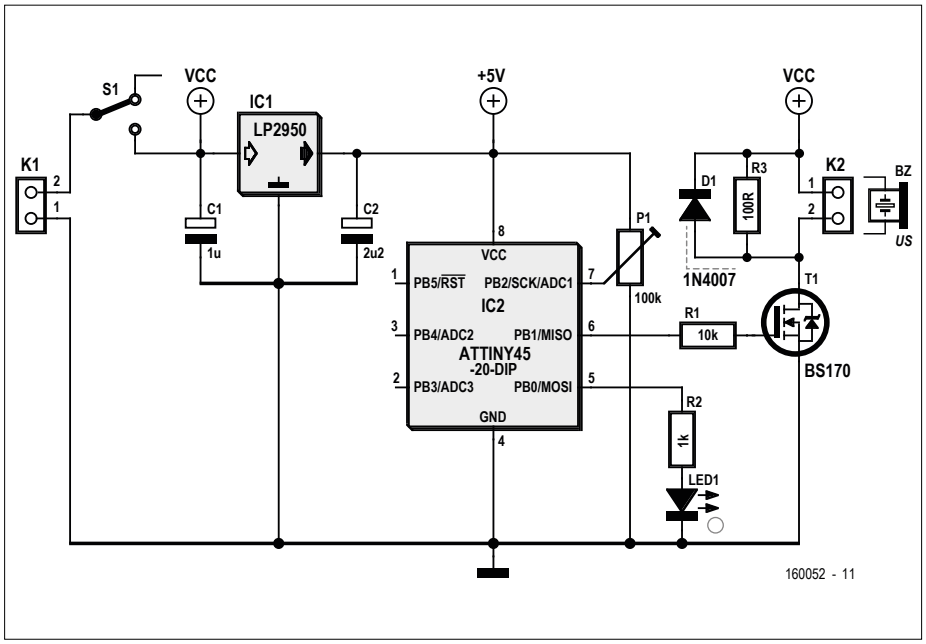



Bild 1. Die Schaltung der Marderscheuche ist klein und einfach, sie kann auch von Einsteigern erfolgreich aufgebaut werden.

dachten wir, eine Marderscheuche kann heute kompakter (und kostengünstiger!) konstruiert werden.
Gesagt, getan – wir machten uns an die Arbeit. Die Wunschliste unseres Lesers war eher bescheiden:

- Anwenderfreundliche Handhabung
- Unkomplizierter Aufbau, keine SMDs
- Einstellbare Frequenz 20...40 kHz
- Hohe, marderwirksame Lautstärke
- Niedrige Kosten
- Energiequelle 9-V-Batterie

## Scheuchen-Schaltung

Die Schaltung unserer Ultraschall-Marderscheuche in **Bild 1** ist ebenfalls bescheiden ausgefallen. Im Mittelpunkt steht, wie könnte es anders sein, ein Mikrocontroller: Der kleine ATtiny25-20 von Atmel. Betrachten wir zuerst die Stromversorgung. Die Betriebsspannung, die eine 9-V-Batterie, der Fahrzeug-Akku oder ein Steckernetzteil liefert, kann im Bereich 9...12 $V_{DC}$ liegen. Von Anschluss K1 gelangt sie über Schalter SW1 zum Low-drop-Spannungsregler IC1, ein LP2950 in der 5-V-Version. Die Kondensatoren C1 und C2 tragen zur Stabilität der Ausgangsspannung bei.
Der Mikrocontroller ATtiny25-20 wird an der stabilisierten Spannung +5 V betrieben. Für die Ausgangsstufe ist eine stabilisierte Spannung entbehrlich, sie liegt deshalb an der unstabilisierten Batterie-, Akku- oder Netzteilspannung.
Der Mikrocontroller (IC2) kommt mit einem Minimum an Peripherie aus. Er ist der Generator der Signale, vor denen Marder (hoffentlich!) Reißaus nehmen. Mit Trimmpoti P1 lässt sich die Frequenz zwischen ungefähr 20 kHz und 43 kHz einstellen. Das Mikrocontroller-Programm ist so geschrieben, dass der Generator ein intermittierendes Signal liefert. Im Abstand von zehn Sekunden wird ein eine Sekunde langer Ultraschall-Ton erzeugt. Dadurch hält nicht nur eine Batterie wesentlich länger als im Dauertonbetrieb durch, auch der Gewöhnung der Marder-Ohren wird vorgebeugt. Da Ultraschall-Töne für Menschen-Ohren nicht hörbar sind, macht die LED1 die Intervalle sichtbar.
Der Ausgang des Mikrocontrollers (Pin 6) ist nicht in der Lage, einen Ultraschallgeber unmittelbar zu steuern. Deshalb wurde die Treiberstufe mit dem MOSFET BS170 (T1) nachgeschaltet.

## Der Tweeter

Das verstärkte Ausgangssignal des Mikrocontrollers muss von einem elektroakustischen Wandler in Ultraschall-Bursts umgesetzt werden, die für das Marder-Ohr möglichst ohrenbetäubend wirken. Diese

Bild 2. Messungen im Elektor-Labor ergaben, dass der Tweeter MPT-001 von Monacor für diesen Einsatz hervorragend geeignet ist.

Aufgabe übernimmt ein so genannter Piezo-Tweeter, ein Hochton-Lautsprecher mit hohem Wirkungsgrad. Nach einigen Probeläufen im Elektor-Labor, allerdings ohne Marder, haben wir uns für das Modell MPT-001 von Monacor entschieden (**Bild 2**). Dieser Hochtöner ist zwar nicht als Ultraschallgeber konzipiert, doch er vereint auch bei Ultraschall einen ansehnlichen Wirkungsgrad mit niedrigen Abmessungen zu einem sympathischen Preis. Falls ein anderes Modell der Favorit ist oder ein Ultraschall-Tweeter im Lager oder in der Bastelkiste auf seinen Einsatz wartet: Es spricht nichts dagegen, damit zu experimentieren.

## Aufbau und Einsatz

Für die Marderscheuche hat das Elektor-Labor eine wenige Quadratzentimeter große Platine entworfen, **Bild 3** zeigt das Layout und die Bestückung. Da ausschließlich bedrahtete (*Through-hole*) Bauelemente verwendet werden, dürften auch Einsteiger den Aufbau in weniger als einer Stunde schaffen. Wir empfehlen, für den Mikrocontroller eine Fassung zu montieren!

Die Inbetriebnahme gestaltet sich einfach: Mit Trimmpoti P1 wird die Frequenz des Ausgangssignals eingestellt. Kontrollieren lässt sich die Frequenz am besten mit einem Oszilloskop oder Frequenzzähler. Wenn kein derartiges Gerät zur Hand ist, lässt sich die Funktion auch „nach Gehör" prüfen. Das eigentliche Signal ist für menschliche Ohren natürlich nicht hörbar, hörbar ist aber (nach dem Einschalten) ein im Abstand von zehn Sekunden aus dem Tweeter kommendes „Tick"-Geräusch. Dies ist ein Indiz dafür, dass die Marderscheuche in Funktion ist.

**Listing 1. Dieses Programm lehrt die Nager das Fürchten.**

```
//Attiny25 , running @ 1MHZ
// Using timer 1
//
//                      +-\/-+
// Ain0  (D  5)  PB5  1|    |8  VCC
// Ain3  (D  3)  PB3  2|    |7  PB2  (D  2)  INT0  Ain1 <- Potmeter input
// Ain2  (D  4)  PB4  3|    |6  PB1  (D  1)        pwm1 <- Speaker output
//               GND  4|    |5  PB0  (D  0)        pwm0 <- Led output
//                      +----+

//  calculate output frequency
//  clockspeed / prescaler / OCR1C = frequency(Hz)
//  clockspeed / prescaler / frequency = OCR1C
//  OCR1C = OCR1A

#include <avr/sleep.h>
#include <avr/interrupt.h>
#include <avr/wdt.h>

#define potmeter A1

#define adcDisable() (ADCSRA &= ~(1<<ADEN)) // disable ADC
#define adcEnable()  (ADCSRA |=  (1<<ADEN)) // re-enable ADC
byte compareValue = 0;

void setup() {
  disableWatchdog();
  adcEnable();
  compareValue = map(analogRead(potmeter), 0, 1023, 25, 10);
                 startTimer1(); //20 - 50 kHz -> OCR0A: 25 - 10
                   adcDisable();
                    pinMode(0, OUTPUT);
                    digitalWrite(0, HIGH);
                    delay(1000);
                    digitalWrite(0, LOW);
                    stopTimer1();
                    enableWatchdog();
                    enterSleep();
```

Schlagen Sie unliebsame Nager effizient in die Flucht!

## Das Programm

Das Mikrocontroller-Programm, das wir in nebenstehendem Kasten zusammengefasst haben, bedarf nur wenig Erklärung. Nach dem Reset wird die Stellung des

```
}

void loop() {
}

void startTimer1(void) {
  pinMode(1, OUTPUT);
  TCNT1 = 0;
  TCCR1 = 0;
  GTCCR |= (1 << PSR1);   //section 13.3.2 reset the prescaler
  TCCR1 |= (1 << CTC1);   // section 12.3.1 CTC mode
  TCCR1 |= (1 << COM1A0); //togle pin PB1 table 12-4
  TCCR1 |= (1 << CS10);   //prescaler 1 table 12-5
  //TCCR1 |= (1 << CS11);
  //TCCR1 |= (1 << CS12);
  //TCCR1 |= (1 << CS13);
  OCR1C = compareValue;
  OCR1A = compareValue;
}

void stopTimer1(void) {
  TCNT1 = 0;
  TCCR1 = 0;
}

void enableWatchdog(void) {
  wdt_enable(WDTO_8S); //enable watchdog
}

void disableWatchdog(void) {
  MCUSR &= ~(1<<WDRF); // reset status flag
  wdt_disable();       //reset watchdog
}

void enterSleep(void) {
  set_sleep_mode(SLEEP_MODE_PWR_DOWN);
  sleep_mode();
}
```

## STÜCKLISTE

**Widerstände (0,25 W):**
R1 = 10 k
R2 = 1 k
R3 = 100 Ω
P1 = Trimmpoti 100 k, liegend

**Kondensatoren:**
C1= 1 µF/16 V
C2 = 2,2 µF/16 V

**Halbleiter:**
IC1 = LP2950 (5 V)
IC2 = ATtiny25-20 DIP (programmiert)
D1 = 1N4007
T1 = BS170
LED1 = LED weiß

**Außerdem**
K1,K2 = Buchsenkontaktleiste 2-polig, RM 2,54 mm
S1 = Schiebeschalter für Platinenmontage
BZ = Tweeter Monacor MPT-001 oder vergleichbar
Platine 160052-1

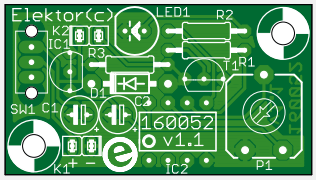


Bild 3. Die einseitige Platine für die Marderscheuche ist im Elektor-Shop erhältlich.

Trimmpotis P1 abgefragt, und der Timer wird konfiguriert. Danach startet die Marderscheuche ihr (hoffentlich segensreiches!) Werk. Natürlich ist das Modifizieren des Programms möglich, dem Experimentieren sind keine Grenzen gesetzt. Variieren lassen sich beispielsweise die Dauer des Signal-Bursts oder die Intervallzeit. Etwas Programmiererfahrung vorausgesetzt ist auch das Schreiben eines anders konzipierten, vielleicht noch effizienteren Programms möglich.
Unsere Firmware kann gratis von der Projektseite [1] heruntergeladen werden. Der Elektor-Shop hat den programmierten Mikrocontroller für wenig Geld in seinem Sortiment. Zum Programmieren in eigener Regie eignet sich der Programmer TL866A, den der Elektor-Shop ebenfalls anbietet [2].

(160052)gd

## Weblinks

[1] www.elektormagazine.com/160052
[2] www.elektor.de/tl866a-universal-programmer

## EINKAUFSZETTEL

→160052-1
Platine

→160052-41
Programmierter Controller

# Bandgap-Referenz

## Bemerkenswerte Bauteile

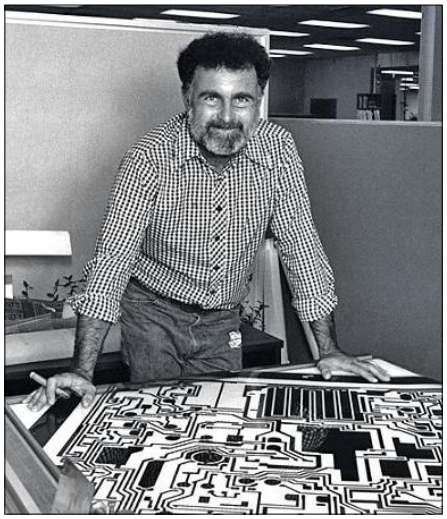


Bild 1. Robert John (Bob) Widlar (1939 bis 1991), berühmter Analog-Entwickler und *enfant terrible* des Silicon Valley.

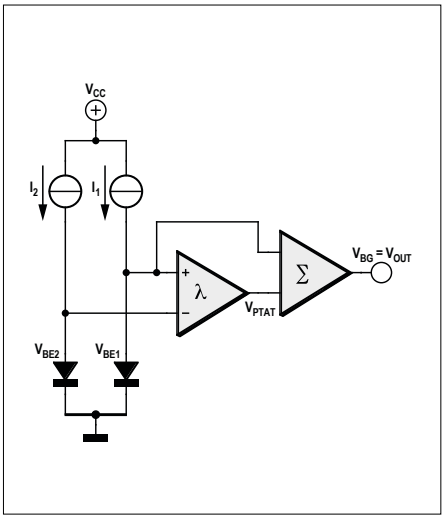



Bild 2. Bandgap-Spannungsreferenz (Quelle: Texas Instruments).

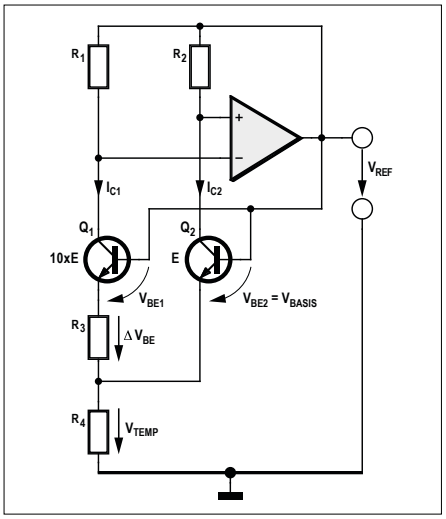



Bild 3. Paul Brokaws verbesserte Bandgap-Referenz.

Von **Neil Gruending** (Kanada)

Spannungsreferenzen sind essentielle Bestandteile von Spannungsreglern, ADCs und einer Vielzahl anderer analoger ICs. Ein spezieller Typ mit dem Vorsatz „Bandgap“ wird häufig als temperaturkompensierte Referenz mit hoher Genauigkeit genutzt. Darüber macht man sich auch als Elektroniker in der Regel wenig Gedanken, denn dieser Teil steckt gut versteckt in den ICs. Doch für die Serie „Bemerkenswerte Bauteile“ ist durchaus ein tieferer Einblick angebracht.

Einer der ersten Wege zu einer konstanten Spannung war die Verwendung einer kleinen Glimmlampe. Nach der Zündung liegt die Spannung über den beiden Elektroden im Bereich von 48... 80 V – abhängig vom Fabrikat. Zum Betrieb genügte ein kleiner Strom, doch wenn einmal die Spannung zu klein wurde, musste sie zuerst wieder über die höhere Zündspannung (66... 200 V) ansteigen, damit das enthaltene Gas wieder ionisiert wird. Mit diesen Einschränkungen war es schon 1966 möglich, eine auf etwa ±0,5 V genaue Brennspannung zu realisieren.
Doch sehr bald wurden Glimmlampen durch Z-Dioden ersetzt. Mit diesen „Shunt-“ bzw. Parallelreglern erzielte man höhere Genauigkeiten, indem man den Spannungsdurchbruch des pn-Übergangs einer „verkehrt“ herum betriebenen Diode ausnutzte. Es gibt sie für sehr viele und auch kleinere Spannungen, was ihren Einsatz sehr viel flexibler machte. In den 1970ern wurden sie sogar schon in ersten Spanungsregler-ICs eingesetzt. Für optimales Verhalten wurde diese Diode im Die „vergraben“, was vor allem das Rauschen reduzierte.

Z-Dioden waren aber aus vielerlei Gründen nicht wirklich ideal für monolithische Spannungsregler-ICs. Bob Widlar (**Bild 1**), der berühmte Analog-Ingenieur von National Semiconductor, behauptete in einem EEE-Artikel sogar, dass monolithische Spannungsregler-ICs damit quasi unmöglich wären, da der integrierte Leistungstransistor mit seiner Wärme einen negativen Temperatur-Effekt auf Die und Z-Diode hätte. Und nur ein Jahr später veröffentlichte Widlar den ersten richtigen monolithischen Spannungsregler: das IC LM109. Sein Ausweg aus den negativen Temperatur-Effekten war die Verwendung der Bandgap-Referenz in **Bild 2**.
Der Trick lag darin, die beiden Dioden mit unterschiedlichen Strömen zu betreiben, um einen sogenannten PTAT-Strom (Proportional-To-Absolute-Temperature) zu generieren. Das Besondere ist, dass sich so die Temperaturkoeffizienten der beiden Dioden gegenseitig kompensieren, was eine temperaturstabile Ausgangsspannung von 1,2 V – die Bandabstandsspannung von Si-Halbleitern – ergab.

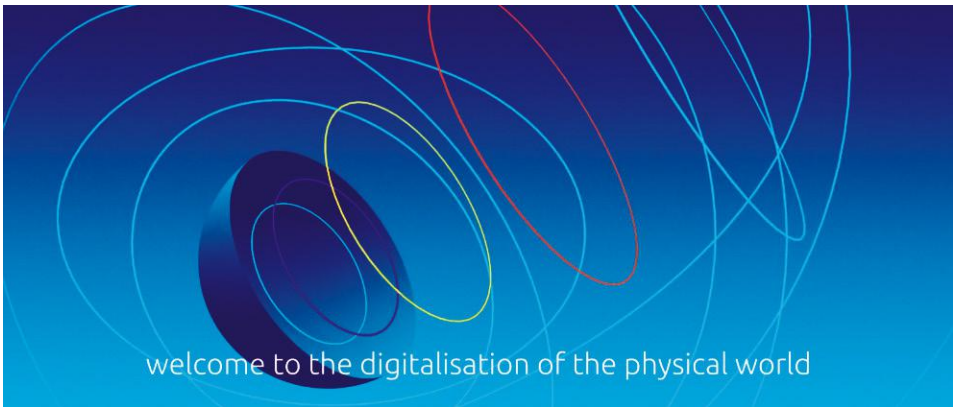
Paul Brokaw schraubte dann daran herum und führte 1974 eine Gegenkopplung ein (**Bild 3**). Seine trickreiche Gegenkopplung sorgt für gleiche Emitterströme, wodurch sich die Temperaturkoeffizienten exakt ausgleichen (wobei die Emitterflächen der beiden Transistoren unterschiedlich sind).
Jetzt wissen Sie also, um was es sich bei Bandgap-Referenzen handelt, wie sie entstanden sind und warum sie so vielfältig eingesetzt werden.

(160074)

**Steuern Sie weitere Bauteile hinzu: Mailen Sie an neil@gruending.net.**

# sigfox Welt der „Things“

Sigfox hat ein Netzwerk aufgebaut, das vielgestaltigen Dingen oder Objekten, heute kurz „Things“ genannt, den Weg ins Netz der Netze bahnt. Gegenwärtig ist Sigfox in mehr als 29 Ländern vertreten. Dort kann sich die IoT-Branche schon heute auf diese zukunftweisende Infrastruktur stützen, zumal die Reihe etablierter Kooperationspartner ständig wächst. Unser Beitrag zeichnet ein kurzes Portrait von Sigfox, anschließend werden Boards und Module für das IoT vorgestellt.




Von **Mariline Thiebaut-Brodier** (Elektor)

## Aktivitäten

Gegründet wurde Sigfox im Jahr 2010 von Ludovic Le Moan und Christophe Fourtet in Labège nahe Toulouse in Frankreich. Das vom Unternehmen entwickelte drahtlose Netzwerk operiert mit niedrigen Übertragungsgeschwindigkeiten in öffentlich freigegebenen Frequenzbändern. Die Technik wurde für die Kommunikation von Maschine zu Maschine (M2M) sowie für den Datenaustausch von IoT-Systemen mit dem Internet konzipiert. Wie schon erwähnt, ist Sigfox in über 29 vornehmlich europäischen Ländern präsent, bis Ende 2018 sollen rund 60 weitere Länder hinzukommen [1]. Sigfox beschränkt sich nicht auf das Installieren eigener Infrastrukturen, sondern ist auch Kooperationspartner internationaler Telekommunikationsdienstleister wie SFR (Frankreich) oder Albertis Telecom (Spanien). Die Kunden dieser Anbieter profitieren neben den angestammten Diensten (4G, WLAN ...) auch von dem Potential, das Sigfox bereitstellt.

Ein anderer erfolgreicher Geschäftszweig ist die Entwicklung netzwerkfähiger Messsysteme, die Verbrauchswerte von Strom, Gas oder Wasser beim Verbraucher vor Ort erfassen. Auch das Überwachen von Verteilnetzen, zum Beispiel das Dokumentieren von Netzlasten oder das Erkennen von Rohrleitungslecks, gehört zu den Aktivitäten dieses Tätigkeitsbereichs. Sigfox pflegt die Zusammenarbeit mit Anbietern von Cloud-Diensten wie Microsoft Azure, um nur einen Namen zu nennen. Die über das Sigfox-Netzwerk angelieferten Daten werden der Cloud zur weiteren Verarbeitung übergeben.

Die Bereiche des täglichen Lebens, die hohen Bedarf an Netzwerkanbindungen haben, wachsen in rasantem Tempo: Infrastrukturen in urbanen Räumen, innovative Logistiksysteme und die Sicherheitstechnik sind nur einige Stichworte. Auch das Orten von Menschen, Tieren oder Objekten (GPS-Tracking) hat stark an Bedeutung gewonnen. Ein Beispiel aus der Ornithologie ist *Chickenrun* von *HidnSeek*, ein System, das Vogelflugwege verfolgt und außer den genauen Positionen auch die Werte von Temperatur und Luftdruck überträgt.

## Drahtlostechnik

Das Sigfox-Netzwerk transportiert begrenzte Informationsinhalte mit niedrigen Geschwindigkeiten (100 bit/s) über weite Distanzen. Der Rekord des überbrückten Abstands zwischen Antenne und Objekt liegt bei 1024 km. Die Weite der Übertragungsstrecke hängt in hohem Maß von der Beschaffenheit des Geländes ab, realisierbar sind normalerweise etwa 2 km bis 130 km. Der Energiebedarf der Objekte ist äußerst gering, denn der Funkverkehr wird in einem ultraschmalen Frequenzband abgewickelt (*Ultra Narrow Band*, UNB). Im Sender moduliert das Nutzsignal einen Träger, indem es die Phase zwischen zwei Werten schaltet (BPSK, *Binary Phase-shift Keying*). Der Empfänger ist für den schmalen Ausschnitt aus dem Frequenzspektrum optimiert. Nach Fehlerprüfung der Daten und eventueller Korrektur gibt der Empfänger die Informationen an die Sigfox-Cloud oder eine vom Auftraggeber vorgegebene Plattform weiter.

Das Sigfox-Netzwerk arbeitet bidirektional, jedoch nur auf Anforderung durch das Objekt. Im Protokoll ist festgelegt, dass das Netzwerk dem Objekt Dienste anbietet, nicht umgekehrt. Das Objekt leitet das Aktualisieren der im Netzwerk zwischengespeicherten Daten ein, und sofern nötig, schickt das Netzwerk eine Antwort zurück. Während der übrigen Zeit ist das Netzwerk inaktiv. Das Energie-Management wurde für den Betrieb an unabhängigen Stromquellen optimiert, denn manche Objekte sind über Zeiträume von zehn Jahren und länger auf einen Satz Batterien angewiesen. Der Energiebedarf von Sigfox-Objekten lässt sich unkompliziert kalkulieren, da die mittlere Stromaufnahme konstant ist. Natürlich hängt der Energiebedarf auch von der Anzahl der übertragenen Datenpakete ab.

| Tabelle 1. Das Sigfox-Netzwerk in den Regionen der Welt. | |
|---|---|
| **Radio-kommunikations-zone** | **Region** |
| **RCZ1** | Europa (868 MHz, ETSI 300-220), Oman, Südafrika |
| **RCZ2** | USA (902 MHz, FCC Part 15), Mexiko, Brasilien |
| **RCZ3** | Japan (923 MHz, ARIB STD-T108) |
| **RCZ4** | Australien, Neuseeland, Singapur, Taiwan, Hongkong, Kolumbien, Argentinien (920 MHz, ANATEL 506, AS/NZS 4268, IDA TS SRD) |

Innerhalb eines Tages kann ein Objekt dem Netzwerk bis zu 140 Datenpakete übergeben. In jedem Datenpaket sind bis zu 12 beliebig nutzbare Datenbits untergebracht.
Sigfox unterscheidet sich von manchen Mitbewerbern nicht nur in der technischen Realisierung, sondern auch im Geschäftsmodell. Da die Sigfox-Technik offen ist, können Halbleiterhersteller das Sigfox-Protokoll frei in ihre Produkte integrieren. Die Rechteinhaber mancher konkurrierender Systeme fordern Lizenzkosten für jeden produzierten Chip. Sigfox verlangt nur Geld für die zu leistenden Dienste, die der Kunde zu seinem Vorteil nutzt. Das Geschäftsmodell ist für die konkurrierenden Hersteller von Funkmodulen ein Anreiz, die Kosten niedrig zu halten. Das preisgünstigste Modul für die Anbindung an das Sigfox-Netzwerk kostet derzeit rund 2 €.
Es ist deutlich, dass sich die Technik des Sigfox-Netzwerks von anderen Netzwerken grundlegend unterscheidet. Die technischen Eigenschaften tragen maßgeblich zur schnellen und weltweiten Verbreitung bei. Wichtige Pluspunkte sind, verglichen mit den Konkurrenten, die hohen Reichweiten sowie die niedrige Anzahl der notwendigen Antennen. Außerdem verfügt Sigfox über enorme System-Kapazitäten, so dass Engpässe auch auf sehr lange Sicht nicht zu befürchten sind.

### Boards und Module

Wenn Sie Erfahrungen mit Sigfox sammeln möchten, brauchen Sie ein Sigfox-Modul oder ein Sigfox-Entwicklungskit. Sie

**Tabelle 2. Integrierte Halbleiterbausteine für die Sigfox-Kommunikation, die in Shields, Entwicklungskits und Erweiterungskarten verbaut sind.**

| Hersteller | Typ | Kategorie | Anwendung |
|---|---|---|---|
| Atmel | ATA8520D-GHQW | Transceiver + 8-bit-AVR-Mikrocontroller | |
| | ATA8520E-GHQW | Transceiver + 8-bit-AVR-Mikrocontroller | Entwicklungskit: ATA8520-EK4-E (RCZ1) / ATA8520-EK1-F (RCZ2); Erweiterungskarte für Xplained Pro: ATA8520-EK6-E (RCZ1) / ATA8520-EK3-F (RCZ2) |
| | ATA8520-GHQW | Sender + 8-bit-AVR-Mikrocontroller | |
| OnSemiconductor | AX-SFEU / AX-SFEU-API | Transceiver | Entwicklungskit: DVK–SFEU–[API]–1–GEVK |
| | AX-SFUS / AX-SFUS-API (RCZ2) | Transceiver | |
| Radiocrafts | RC1682-SIG | Radiomodul (Controller + Sigfox-Stack) | Demokit: RC1682-SIG-DK |
| STMicroelectronics | S2-LP | Transceiver | Entwicklungskit: STEVAL-FKI868V1 oder STEVAL-FKI915V1 (Board: NUCLEO-L152RE + S2-LP) |
| TD Next | TD1204 | Sigfox-Sender + GPS-Empfänger | Entwicklungsboard: TD1204 EVB |
| | TD1205P | Sigfox-Sender + GPS-Empfänger + Antenne + Beschleunigungssensor | |
| | TD1207R/08R | Gateway (Sender + ARM-Cortex-M3-Mikrocontroller) | Entwicklungsboard: LGA25 EVB |
| | TD1508 (RCZ2) | Gateway (Sender + ARM-Cortex-M3-Mikrocontroller) | Entwicklungsboard: LGA25 EVB |
| Telit | LE51-868 S | Transceiver | Entwicklungskit: LE51-868S EVK |
| Texas Instruments | CC1120 / CC1125 / CC1190 | Transceiver | Entwicklungskit: MSP430F5529 LaunchPad oder CC1120 CC1190 BoosterPack oder CC1125 BoosterPack |
| Wisol | WSSFM10R1 | Transceiver | Erweiterungskarte: BRKWS01 von SNOC |

**Tabelle 3. Verbindungsklassen abhängig von der Strahlungsleistung (vom Objekt zum Netzwerk).**

| | Klasse U0 | Klasse U1 | Klasse U2 | Klasse U3 |
|---|---|---|---|---|
| **ETSI** | 14 dBm ±2 dB | 12 dBm > P > 7 dBm | 7 dBm > P > 0 dBm | < 0 dBm |
| **FCC** | 22 dBm ±2 dB | 20 dBm > P > 15 dBm | 15 dBm > P > 5 dBm | < 5 dBm |

müssen sich für eine Weltregion entscheiden, in der das System betrieben werden soll. Die regionalen Unterschiede betreffen die Regelungen und Normen, die für drahtlose Übertragungsverfahren gelten, sie sind in **Tabelle 1** zusammengefasst. Sie brauchen einen für Sigfox geeigneten Transceiver oder ein Übertragungsmodul sowie einen Mikrocontroller. Aus **Tabelle 2** gehen die erforderlichen Komponenten hervor. Die Module werden entweder über eine serielle Schnittstelle (UART) mit AT-Kommandos gesteuert (Kommunikation auf niedriger Ebene), oder über eine API mithilfe von Bibliotheken.

Wenn Sie zu den fortgeschrittenen Anwendern eines Arduino-Boards oder eines Raspberry Pis gehören, wird es nicht schwierig sein, für Ihr System die passende Sigfox-Erweiterung zu finden. Falls das nicht zutrifft, können Sie auf ein plattformunabhängiges Sigfox-Entwicklungsboard ausweichen. Wir stellen Ihnen nachfolgend eine Auswahl marktgängiger Boards oder Module vor. Die meisten tragen das Label *Sigfox Ready*, das von Sigfox nach einer kostenpflichtigen Prüfung (2500 €) vergeben wird. Dabei wird auch die Strahlungsleistung gemessen, sie ist maßgeblich für das Zuordnen zu den auf ETSI- oder FCC-Empfehlungen beruhenden Klassifizierungen (siehe **Tabelle 3**). Die Klasse U0 ist ein Indiz für die höchste Reichweite.

Im Preis eines Kits ist meistens der Zugang zum Sigfox-Netzwerk für die Dauer eines Jahres enthalten, so dass einem spontanen Einstieg in die Sigfox-Welt nichts im Weg steht. Die Folgekosten bewegen sich zwischen 1 € und 20 € im Jahr. Sigfox unterstützt Startup-Unternehmen, die Produkte an das Netzwerk anbinden möchten, durch hohe Rabatte und kostenfreie Tools. Das Programm *Startup* ist speziell darauf zugeschnitten, die Entwicklung neuer Produkte zu beschleunigen.

Wenn Sie ein Entwicklungskit oder ein System-on-Chip (SoC) ohne Registrierung auf einer Plattform des Herstellers erwerben, müssen Sie Ihr Objekt bei einer Sigfox-Plattform anmelden (siehe Liste auf der Website [2]). Manche Plattformen sind für spezielle Anwendungen reserviert, andere wie Microsoft IoT, Amazon AWS oder Thethings.io stehen allen Anwendungen offen.

Sigfox-Objekte lassen sich anhand folgender Kriterien identifizieren:

- Eine eindeutige SIGFOX ID (*Read-only Device Unique Identifier*), sie ist meistens auf einem aufgeklebten Etikett vermerkt. Diese ID, zum Beispiel *16B87*, ist nicht modifizierbar, sie wird zum Registrieren im Netzwerk benötigt und darf nicht verlorengehen.
- Ein *Porting Authorization Code* (PAC), er ist ebenfalls zum Registrieren erforderlich. Dieser Code im Format *CF14xxxx-xxx* ändert sich, wenn das Objekt den Eigentümer wechselt.
- Ein versteckter Sicherungscode, verschlüsselt mit AES128.

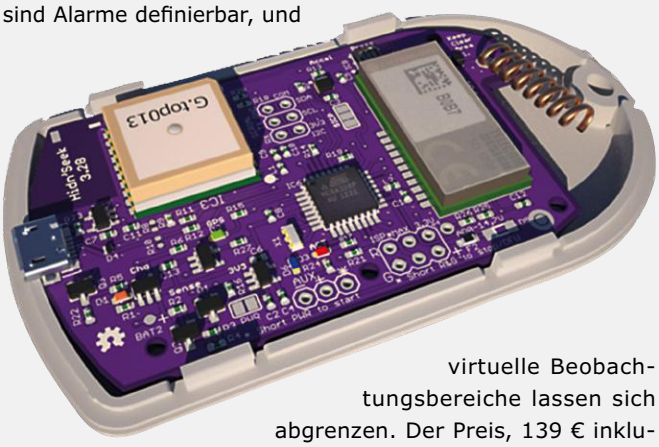
Die Palette der hier vorgestellten Produkte lässt erahnen, dass das Angebot für den Zugang zu Sigfox umfassend und vielfältig ist. Hinzugefügt sei noch, dass Sigfox eine Website für Entwickler pflegt [3], sie enthält Tutorials und weiterführende Weblinks.

Wir fassen zusammen: Das Entwickeln von Objekten, die an das „Internet of Things“ angebunden sind, wird durch niedrigpreisige, unkompliziert beschaffbare Module unterstützt. Der Zugang zum Sigfox-Netzwerk lässt sich einfacher und kostengünstiger herstellen als beispielsweise zu LoRa. Dem Realisieren Ihres ersten Objekts steht nichts entgegen. Ob Sie den Wegen Ihres streunenden Vierbeiners nachspüren wollen oder Ihre Ambitionen darin liegen, Umweltgrößen über weite Distanzen hinweg zu messen: Mit Sigfox ist alles möglich! Wenn Sie Sigfox in ein allgemein interessierendes Projekt einbauen, würden wir uns freuen, davon zu erfahren. ◀

(160167)gd

## HidnSeek – kompatibel mit der Arduino-IDE

HIDNSEEK ST1-A/B ist die Typenbezeichnung eines autonomen, wiederaufladbaren GPS-Trackers ohne SIM-Karte von HidnSeek [4]. Mit einer mobilen App können auf einer Webseite mehrere Tracker verfolgt werden. Außerdem sind Alarme definierbar, und



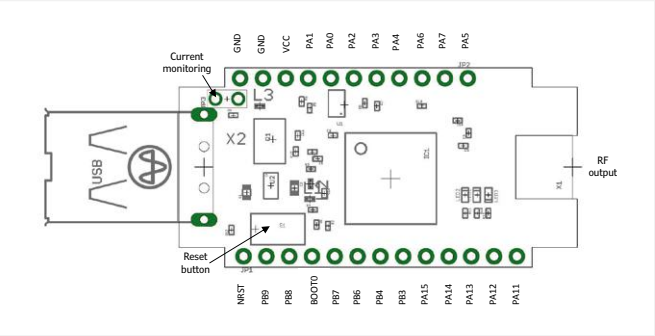
virtuelle Beobachtungsbereiche lassen sich abgrenzen. Der Preis, 139 € inklusive Mehrwertsteuer, schließt die Anbindung an das Sigfox-Netzwerk für ein Jahr ein, der Vertrag ist für jährlich 19,90 € verlängerbar.

Der Tracker von HidnSeek hebt sich dadurch hervor, dass sowohl die API als auch die Firmware offen sind. Damit wird er zum Entwicklungsboard, das durch Hinzufügen von Sensoren individuell anpassbar ist. Der Hersteller hat auf GitHub [5] die Tools bereitgestellt [6], die zum Programmieren über die Arduino-IDE und ein USB-Kabel nötig sind. Auf dem Board befinden sich ein mit 12 MHz getakteter Mikrocontroller ATmega328P, ein Sigfox-Gateway TD1207 sowie ein GPS-Modul PA6H von GlobalTop.

### Eigenschaften

- Lokalisierung mit GPS
- Dreiachsiger Beschleunigungssensor
- Sensoren für Temperatur und Druck
- Updaten der Firmware über USB
- *Geofencing* mit Apple iOS- und Android-Apps
- Freie GPIOs für Erweiterungen ($I^2C$, UART, ISP, analoge und digitale I/O)
- LiPo-Akku 500 mAh, überwachtes Laden (Betriebsdauer über Monate)

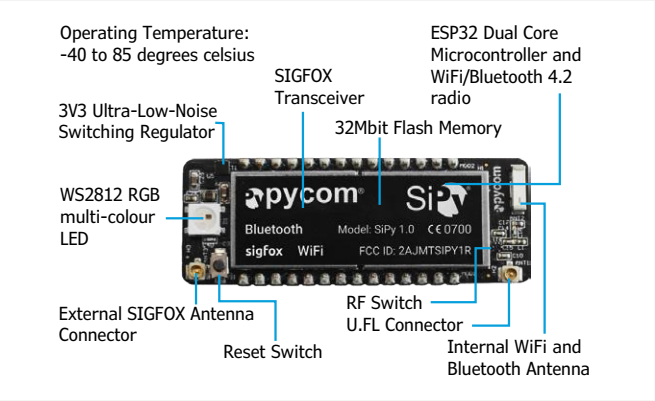
## Nemeus – USB-Stick




Von Nemeus kommt der USB-Stick MM002-LS-EU [7], ein zweifaches Modem (LoRa/Sigfox) mit dem Mikrocontroller STM32L151X und dem Transceiver RF SX1272. Der Stick ist Teil des Steuerungssystems *Contiki*. Die Software kann über eine JTAG-Schnittstelle in den Mikrocontroller geladen werden. Für die Kommunikation mit den Sensoren sind diverse Optionen vorhanden: I²C, SPI, UART, GPIO, ADC, DAC.
Der schnelle Start in das Sigfox-Netzwerk ist mit der Ausführung USB MK002-LS-EU möglich [8]. Dieser Stick kann auf direktem Weg mit einem Windows- oder Linux-Computer verbunden werden, er wird über AT-Kommandos gesteuert. Mit der zugehörigen Java-App lässt sich der Stick testen. Die Stromversorgung ist auch über einen Vcc-Anschluss möglich.

Inzwischen kündigte Nemeus den IoT Smart Sensor an, ein Entwicklungsboard für drahtlose Netzwerkanbindungen (Sigfox, LoRa, Bluetooth, GPS/GNSS…). Dieses Board ist kompatibel mit der Arduino-IDE, auf dem Board befinden sich ein Beschleunigungssensor, ein Temperatursensor, ein Drucksensor und eine vom Anwender programmierbare RGB-LED.

## PyCom – Programmieren in Python



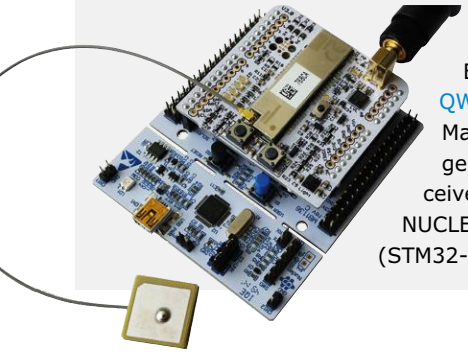

Ein Produkt von PyCom ist das Entwicklungsboard **SiPy** MicroPython Wi-Fi/Bluetooth/Sigfox [9]. Das Board ist mit Micropython kompatibel, es wurde speziell für IoT-Anwendungen entwickelt. Auf dem Mini-Board (55 · 20 · 3,5 mm) befindet sich ein SoC mit Dual-core-Mikrocontroller des Typs ESP32 von Espressif. Das Board ist mit WLAN, Bluetooth (Normal und BLE), Sigfox sowie 512 kB RAM ausgestattet. Programmiert wird mit einer eigenen IDE von PyCom, genannt *Pymakr*.
**Achtung:** Für Sigfox ist eine externe Antenne erforderlich (RS Components 125-9535)! Die Inbetriebnahme ohne Antenne kann Beschädigungen zur Folge haben. Die Antenne für WLAN und Bluetooth ist eingebaut. Die Anschlüsse des Boards sind leicht zugänglich, wenn die universelle Erweiterungskarte von PyCom zum Einsatz kommt. Darauf befinden sich ein Micro-USB-Anschluss für die Stromversorgung und die serielle Kommunikation, ein LiPo-Akkulader, eine Sicherung gegen Verpolen, eine Micro-SD-Speicherkartenfassung, ein Drucktaster und eine Anwender-LED.

### Eigenschaften

- 2 · UART
- 2 · SPI
- I²C
- I²S
- Micro-SD-Karte
- ADC: 8 Kanäle, 12 Bit
- Bis zu 24 GPIOs
- RGB-LED WS2812
- Schalter für RESET und RF
- Transceiver CC1125 von TI
- Versionen: RCZ1/RCZ3 und RCZ2/RCZ4

PyCom hat bereits die Version FiPy WiFi/Bluetooth/LTE CAT M1 / NB1/LoRa/Sigfox angekündigt [10], die bei identischen Abmessungen die Zugänge zu fünf Netzwerk-Systemen herstellen kann.

## Quicksand – Entwicklungsboard mit STM32



Die belgische Firma Quicksand hat das Entwicklungskit QW GPS Sigfox auf den Markt gebracht [11], aufgebaut mit einem Transceiver TD1204 und einem NUCLEO-L152RE-Board (STM32-mbed-Mikrocontroller).

### Eigenschaften

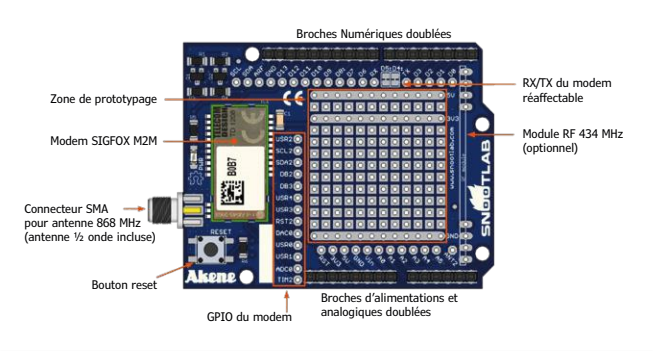
- Sensoren: GPS, Beschleunigung, Temperatur, Näherung und Umgebungslicht
- 2 frei verwendbare Taster, 1 Reset-Taster, 4 frei verwendbare LEDs
- Kompatibel mit Arduino über Level-Shifter
- Kompatibel mit 3,3 V und 5 V
- Open-source-Software verfügbar
- Schnittstelle JTAG nach TD1204

## ATIM – umfassende Auswahl

| Typ | Adapter | Eingang | Ausgang (Antenne) |
|---|---|---|---|
| ACW-DUINO | Arduino | GPIO | SMA- oder UFL-Connector |
| ACW-MangOH | Computer/MangOH | GPIO, UART | UFL-Connector |
| ACW-MPCIE | Mini PCI-Express | UART | UFL-Connector |
| ACW-RPI | Raspberry Pi | GPIO | SMA- oder UFL-Connector |
| ACW-SDK | USB (*Starterkit*) | | SMA-Connector |
| ACW-USB | USB-Stick | USB | SMA-Connector |
| SigBee | XBee | GPIO | SMA- oder UFL-Connector |
| ARM-N8-SIGFOX | Sigfox-Version des drahtlosen ARM-Nano-Moduls (Advanced Radio Modem®) | | |

Der in Frankreich ansässige Hersteller ATIM hat eine ganze Reihe drahtloser Multimodus-Erweiterungsplatinen für Sigfox und LoRa im Angebot [12]. Die nachstehende Tabelle gibt eine unvollständige Übersicht. Die meisten Produkte sind an einen Jahresvertrag mit dem Netzwerk gekoppelt.

## SNOOTLAB – Arduino und Raspberry Pi




Boards mit dem Gateway TD1208 und zugehörigen Bibliotheken sind eine Spezialität von SnootLab [13]:

Akeru 3.3: Kompatibel mit der Arduino- oder Genuino-Umgebung. Das Board ist auch Bestandteil eines Kits, das diverse Sensoren einschließt.
Akene: *Shield* für Arduino oder Genuino.
Breakout TD1208: Die Karte fügt Projekten Sigfox-Konnektivität hinzu, ohne dass der Zugang zu den Gateway-Signalen durch eine gesonderte Schaltung hergestellt werden muss.
Foquinha-Pi: *Shield* für Raspberry Pi (alle Varianten).

Sämtliche Boards sind mit Jahresverträgen für das Netzwerk ausgestattet.

## NetTrotter – 100 % kompatibel mit Arduino

Aus Italien stammt das BiB, das Basic IoT Board [14]: Ein Board, das sich den Wünschen des Anwenders anpasst (Betrieb mit Sensoren für Temperatur und Feuchte, Auslesen von NFC-Tags, Bluetooth Low Energy Communication...). Da das Board mit Arduino absolut kompatibel ist, lässt es sich mit zahllosen Arduino-Shields kombinieren, es ist über die Arduino-IDE programmierbar. Auf dem Board haben ein GPS-Empfänger, ein Beschleunigungssensor, ein Magnetometer und ein Gyroskop ihren Platz.

### Weblinks


[1] Sigfox (Abdeckung, Partner): www.sigfox.com, http://partners.sigfox.com

[2] Sigfox-Plattform zum Registrieren von Objekten: http://partners.sigfox.com/products/platform

[3] Sigfox-Website für Entwickler: http://makers.sigfox.com

[4] GPS-Tracker HIDNSEEK ST1-A: www.hidnseek.fr/hidnseek_st1a

[5] GitHub für HidnSeek: http://github.com/hidnseek/hidnseek

[6] Tutorial für HidnSeek: http://github.com/Bucknalla/sigfox-hidnseek-tutorial

[7] MM002-LS-EU: www.nemeus.fr/en/nemeus-mm002-2

[8] USB MK002-LS-EU: www.nemeus.fr/en/mk002-usb-key

[9] SiPy: www.pycom.io/product/sipy

[10] FiPy: www.pycom.io/product/fipy/

[11] QW GPS Sigfox: http://lpwan.be/wp/product/qw-gps-sigfox-development-kit

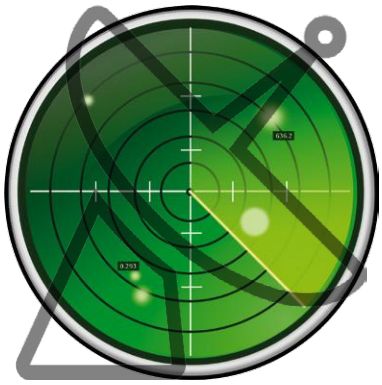
[12] ACW-DUINO, ACW-MangOH, ACW-MPCIE, ACW-RPI, ACW-SDK, ACW-USB, SigBee, ARM-N8-SIGFOX: www.atim.com/en/products/catalog

[13] Akeru 3.3, Akene, Breakout TD1208, Foquinha-Pi: http://snootlab.com/lang-en/72-03-iot-et-sans-fil

[14] BiB, Basic IoT Board: www.nettrotter.io/index.php/it/ecosystem-it/basic-iot-board

# Digitale Korrelation

## Erkennen von Schwingungsform und Periode trotz Rauschen



Von **Robert Lacoste** (F)

Nach Veröffentlichung meiner Beiträge zu den Digitalen Filtern (Elektor Dezember 2016 und Januar 2017) fragte ein Leser an, ob ich auch zum Thema „Korrelationsalgorithmen“ einen einführenden Beitrag schreiben könnte. Dieser Wunsch wird hier erfüllt. Zögern Sie also nicht, den Elektor-Autoren Ihre Wünsche mitzuteilen!

Nehmen wir an, ein stark verrauschtes Signal enthält zu unterschiedlichen Zeiten ein bereits bekanntes Signalmuster. Die Amplitude kann variieren, die Zeiten können sich verschieben, oder das Signalmuster versinkt im Rauschen (Bild 1). Das gesuchte Signalmuster ist zwar bekannt, aber wie können wir das Muster lokalisieren und die Amplitude feststellen? Das scheint nicht einfach zu sein, doch mit der mathematischen Operation „Korrelation“ gelingt dies ohne viel Aufwand.

Bevor wir diese mathematische Operation betrachten, soll zuerst eine typische Anwendung genannt werden: Radar. Ein Radarsystem strahlt über die Antenne ein impulsförmiges Signal ab und versucht sofort anschließend, das von Objekten in der Umgebung reflektierte Signal zu erkennen. Unproblematisch ist das nur, solange die Amplitude des Echos stark ist. Doch wenn intensives Rauschen das Echo überlagert, ist der Empfang schwierig. Trotzdem können heutige Radarsysteme durch Korrelation auch kleine Objekte erkennen, die viele Dutzend Kilometer entfernt sind. Diese Radarsysteme senden nicht nur einen einzelnen Impuls aus, sondern ein komplexes Impulsmuster, das nach digitaler Signalverarbeitung im Echo wiedererkennbar ist.

Die Berechnung einer Korrelation ist nicht schwierig, sie ist auch mit einem Mikrocontroller, DSP oder FPGA leicht durchführbar. Wir betrachten ein Signal, das von einem Analog-Digital-Wandler digitalisiert wurde und in Form einer einspaltigen Tabelle mit N aufeinander folgenden Werten abgelegt ist. Eine zweite einspaltige Tabelle enthält M Werte, die das gesuchte Signalmuster abbilden. Da die zweite Tabelle viel kürzer als die erste Tabelle ist, gilt M < N. Zuerst legen wir die M Elemente des zu suchenden Musters an die ersten M Elemente des zu untersuchenden Signals. Nun werden einzeln an allen M Positionen die Werte der ersten Tabelle mit den Werten der zweiten Tabelle multipliziert, anschließend werden alle Ergebnisse der Multiplikationen miteinander addiert. Das Ergebnis ist der erste Wert der Korrelation.

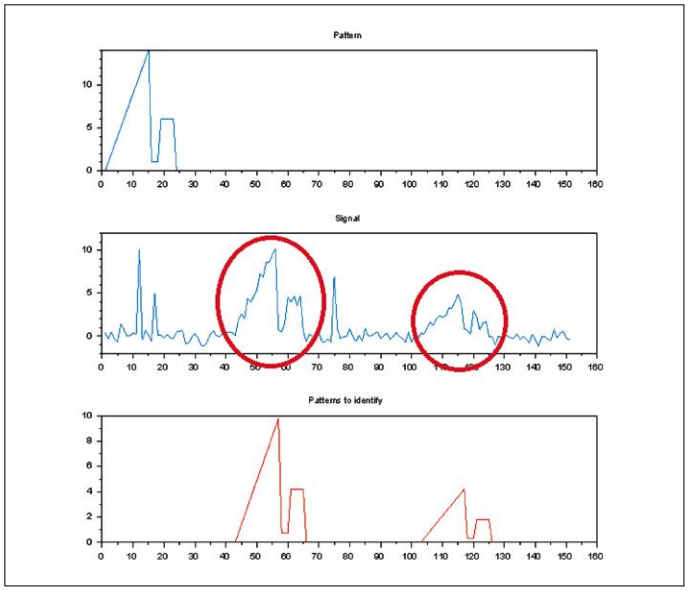



Bild 1. Ein Signalmuster (oben) kommt mit unterschiedlichen Amplituden zweimal in einem gestörten Signal vor (Mitte). Das Signalmuster wird in dem gestörten Signal erkannt (unten).

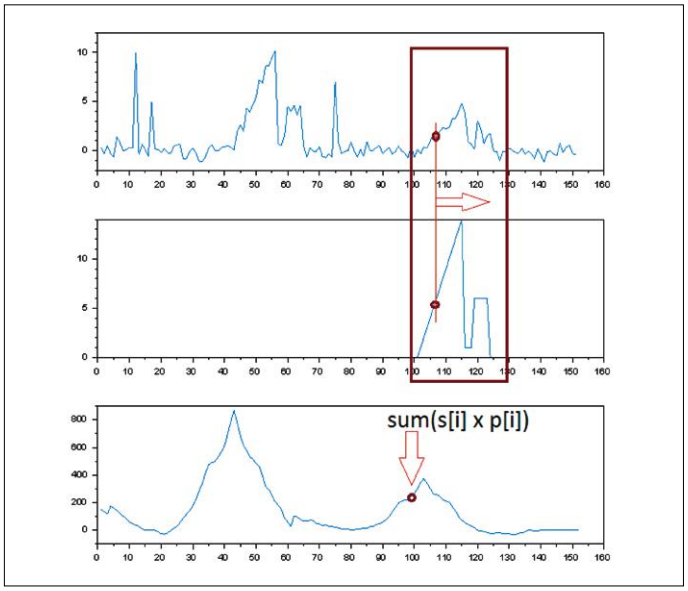



Bild 2. Die übereinander liegenden Werte von Signal (oben) und Signalmuster (Mitte) werden multipliziert. Die Ergebnisse aller Multiplikationen werden addiert. Dies ergibt den ersten Wert der Korrelation (unten). Das Signalmuster wird schrittweise verschoben, und der Algorithmus wird bei jedem Schritt wiederholt.

Im nächsten Schritt wird die Tabelle um eine Position verschoben, in der die M Werte des Signalmusters stehen. Das Multiplizieren und anschließende Addieren wird mit den nun nebeneinander stehenden Werten wiederholt, um den zweiten Wert der Korrelation zu finden. Die Prozedur wird so oft durchgeführt, bis das Ende des zu untersuchenden Signals erreicht ist. Damit ist die Korrelation zwischen diesem Signal und dem Signalmuster berechnet, sie besteht hier aus N – M + 1 Werten. Wenn der Tabellen-Index mit 0 beginnt (wie in der Sprache C), sieht der zugehörige Pseudo-Code wie folgt aus:

```
for i=0 to N-M do
{
 convolution[i]=0
 for j=0 to M-1 do
 {
  convolution[i] = convolution[i] + pattern[j] *
  signal[i+j]
 }
}
```

Der Algorithmus wird in Bild 2 veranschaulicht. Die Frage ist nun, wie die Berechnung der Korrelation die gestellte Aufgabe löst. Das ist einfach: Die Werte des Korrelationssignals steigen mit der Ähnlichkeit zwischen dem zu untersuchenden Signal und dem Signalmuster. Wir müssen folglich die Spitzen des Korrelationssignals betrachten. Ihre Positionen über der Zeit sind ein Indiz für das wahrscheinliche Auftreten des Signalmusters, und ihre Amplituden sind zu den Amplituden des im Signal vorkommenden Signalmusters proportional. In Bild 2 ist dieser Zusammenhang deutlich erkennbar.

## Einige Beispiele

Die Praxis ist natürlich etwas schwieriger als unser erstes Beispiel, denn reale Signale sind viel stärker verrauscht. Trotzdem führt der vorstehende Algorithmus immer noch zu gut brauchbaren Ergebnissen. Um dies zu demonstrieren, habe ich einige Simulationen mit unterschiedlichen Signalmustern und Rauschniveaus dokumentiert. Das Werkzeug, das ich dazu benutzt habe, war das numerische Rechen-Tool Scilab. Wenn Sie die Simulationen am eigenen Rechner nachvollziehen möchten, können Sie dieses Tool kostenlos aus dem Netz [2] herunterladen. Beginnen wir mit einem simplen, rechteckförmigen Signalmuster. Im zu untersuchenden Signal sind zwei Kopien dieses Musters enthalten, das Rauschniveau ist hoch. Damit lautet der Scilab-Code wie folgt:

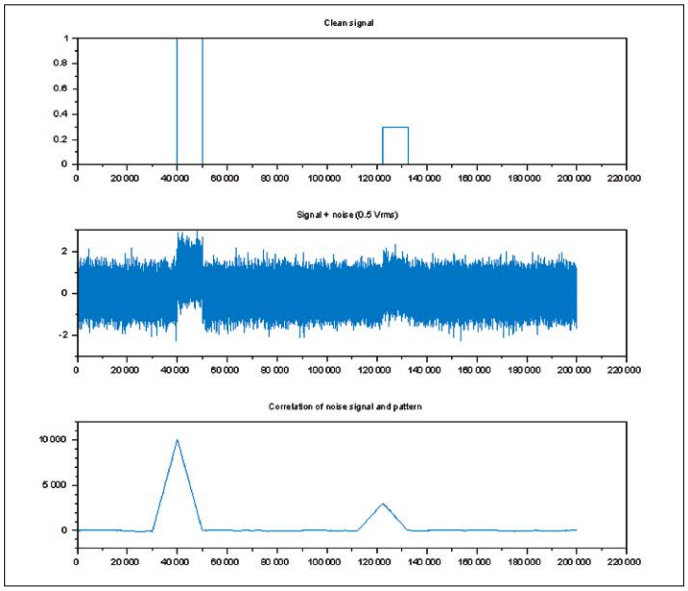



Bild 3. Die Korrelation mit einem rechteckförmigen Signalmuster ergibt die Form eines Dreiecks. Hier besteht das Signalmuster aus 10.000 Werten, es wird deshalb trotz starkem Rauschen sicher erkannt.

```
// Gesuchtes Signalmuster, Rechteck
PatternLength=10000;
Pattern=ones(1,PatternLength);

// Eingangssignal mit zwei Kopien des Signalmusters
SignalLength=200001;
Signal=[1:SignalLength]*0;
PatternPosition1=40000; PatternAmplitude1=1;
Signal(PatternPosition1:PatternPosition1+PatternLeng
   th-1)=Pattern*PatternAmplitude1;
PatternPosition2=122300; PatternAmplitude2=0.5;
Signal(PatternPosition2:PatternPosition2+PatternLeng
   th-1)=Pattern*PatternAmplitude2;

// Hinzufügen von Gaußschem Rauschen
NoiseLevel=0.5;
NoisySignal=Signal+grand(1,SignalLength,"nor",0,NoiseLe
   vel);

// Berechnen der Korrelation (dafür gibt es in Scilab
   eine Funktion...)
Correlation = xcorr(NoisySignal, Pattern);
plot(Correlation);
```

Das Ergebnis dieser Simulation ist in Bild 3 dargestellt: Trotz des Rauschens werden durch Berechnen der Korrelation die Positionen und Amplituden des Signalmusters problemlos erkannt. Übrigens ergibt die Korrelation eines Rechtecks ein Dreieck, weil sie proportional zur Fläche des überlappenden Teils von Signalmuster und Signal ist.

Das Signalmuster in Bild 3 besteht aus 10.000 Abtastwerten. Wenn das Signalmuster nur durch ein Zehntel dieser Werte abgebildet wird, sieht die Simulation wie in Bild 4 aus. Der Algorithmus führt immer noch zum korrekten Ergebnis, die Korrelation enthält jedoch deutlich mehr Rauschen. Wie so oft in der digitalen Signaltechnik bedarf es weniger Anstrengungen, aus dem Rauschen ein Signal zum Vorschein zu bringen, wenn die Anzahl der Abtastwerte höher ist. Hohe Abtastwerte wirken sich einerseits günstig auf das Ergebnis aus, andererseits sind wesentlich mehr Rechenoperationen nötig.

Als Nächstes wollen wir Signale betrachten, die komplexer als ein simples Rechteck sind. Bild 5 zeigt eine Simulation, bei der das Signalmuster einer Periode eines sinusförmigen Signals entspricht. Das Rauschen hat ein so hohes Niveau, dass das Signal fast vollständig untergeht. Auch diesmal liefert der Algorithmus ein korrektes Ergebnis, vorausgesetzt, das Signalmuster besteht aus genügend vielen Abtastwerten (auch hier 10.000). Die Mathematik hat sogar herausgefunden, dass das Berechnen der Korrelation der stärkste lineare Algorithmus ist, um ein vorgegebenes Signalmuster in einem Signal mit weißem Rauschen zu erkennen. Ein System, das durch Berechnen der Korrelation ein bekanntes Signalmuster in einem gestörten Signal erkennt, wird deshalb auch Optimalfilter oder Matched Filter genannt.

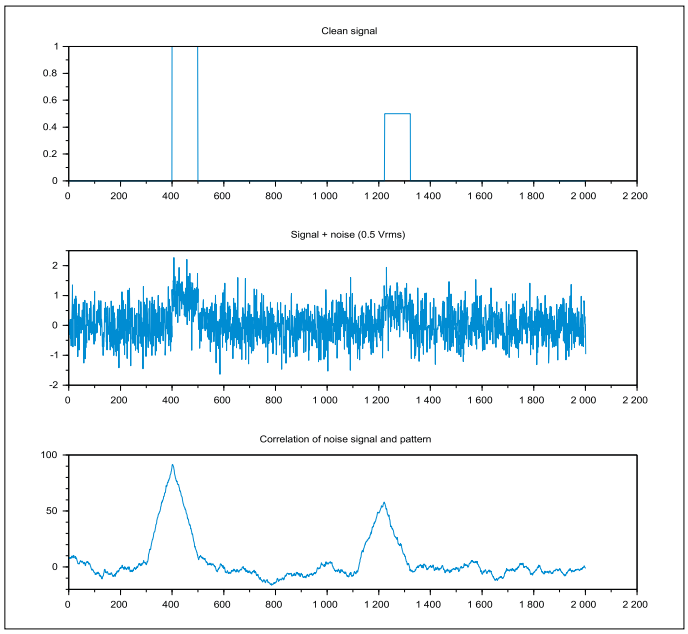



Bild 4. Gleiche Signale wie in Bild 3, das Signalmuster besteht jedoch aus nur 100 Werten. Das Ergebnis ist eine geringere Genauigkeit.

## Bekannt, aber willkürlich

Wir wollen noch einen Schritt weiter gehen. Die Effizienz des Algorithmus steigt mit der Komplexität des Signalmusters, denn die Wahrscheinlichkeit ist niedrig, dass bereits Signalstörungen mit dem komplexen Signalmuster übereinstimmen. Bild 6 zeigt ein eindrucksvolles Beispiel. Das Signalmuster ist ein Pseudo-Random-Signal, das aus 200 willkürlichen Abtastwerten besteht. Das zweimal mit unterschiedlichen Amplituden vorhandene Signalmuster wird von starkem weißem Rauschen überlagert. Im Rauschen ist das Pseudo-Random-Signalmuster nicht erkennbar. Trotzdem ist das Ergebnis der Korrelation überzeugend: Das Vorhandensein des Signalmusters im verrauschten Signal ist im Korrelationssignal durch schmale Impulsspitzen gekennzeichnet. Bei einem Pseudo-Random-Signalmuster

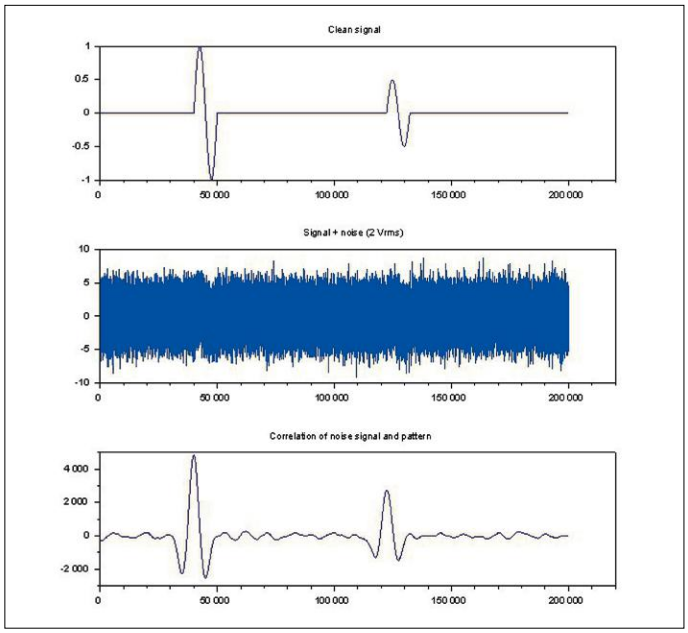



Bild 5. In diesem Beispiel ist das Signalmuster ein sinusförmiges Signal.

besteht nur Korrelation, solange das zu untersuchende Signal mit dem Muster genau übereinstimmt. Daraus resultiert die schmale Impulsspitze, der Rest des Signals wird größtenteils unterdrückt. Der Ort des Musters im Signal lässt sich durch die Korrelation genau feststellen, und das Signal-Rausch-Verhältnis wird effizient gesteigert.

Die Korrelation des Pseudo-Random-Signalmusters führt zu einer effizienten Methode, die Eigenschaften eines Kommunikationssystems zu verbessern. Angenommen, dass über einen störungsbehafteten Kanal ein Bitstrom übertragen werden soll. Wir definieren zwei unterschiedliche Pseudo-Random-Signalmuster, eins für High und eins für Low. Anschließend modulieren wir jedes Bit abhängig von seinem logischen Wert mit einem dieser Muster. Auf der Empfangsseite führen wir die Korrelation des empfangenen Signals mit den beiden Signalmustern durch. Das originale Signal ist nun mühelos rekonstruierbar, selbst dann, wenn das Rauschniveau extrem hoch ist. Das Verfahren ist als Modulationstechnik unter der Bezeichnung DSSS (Direct Sequence Spread Spectrum) bekannt, es wird zum Beispiel bei Zigbee und GPS sowie bei WiFi 802.11b eingesetzt.

Mit diesem Verfahren lassen sich auch mehrere Übertragungskanäle über eine einzige Funkverbindung realisieren, ohne dass sich die Übertragungskanäle gegenseitig stören. Das geht ganz einfach, indem für jeden Übertragungskanal ein anderes Paar aus Pseudo-Random-Signalmustern definiert wird. Die jeweils nicht korrelierenden Übertragungskanäle werden als Rauschen interpretiert. In der Modulationstechnik nennt sich dieses Verfahren CDMA (Code Division Multiple Access), mit ihm arbeitet beispielsweise ein Smartphone im 3G-Netz. Nebenbei ist das Verfahren auch gut geeignet, drahtlose Kommunikation gegen nicht autorisiertes Abhören zu sichern.

## Was muss ein digitaler Prozessor leisten?

Nehmen wir an, dass das Signalmuster aus 100 Abtastwerten besteht (M = 100) und dass in Echtzeit ein Signal verarbeitet werden soll, das mit 10 kS/s abgetastet wird (10.000 Werte pro Sekunde). Zum Berechnen der Korrelation sind für jeden Abtastwert 100 Multiplikationen und Additionen nötig, also 100 · 10.000 = 1 Million derartige Rechenoperationen, und das in jeder Sekunde. Dies lässt sich mit einem über 30 Jahre alten 8-Bit-Prozessor 8051 zwar nicht mehr bewältigen, aber heutige 32-Bit-Prozessoren haben damit kein Problem. Außerdem ist in den meisten ARM-Kernen (wie Cortex M3) und Digitalen Signalprozessoren (DSPs) die Instruktion MAC (Multiply and Accumulate) eingebaut, die eine Multiplikation und Addition in einem einzigen Maschinenzyklus ausführt.

Doch was geschieht, wenn die Anforderungen steigen, beispielsweise weil die Anzahl der Signalmuster-Werte um ein Vielfaches höher liegt? Dann ist ein vielfach schnellerer Prozessor nötig, oder ein FPGA – oder ein mathematisch ausgeklügelteres Verfahren! Es existiert eine Methode, die Korrelationen erheblich effizienter zu berechnen als mit dem vorstehend beschriebenen Algorithmus, sie verwendet zwei FFTs (FFT = Fast Fourier Transformation). Um dies zu erklären, bedarf es eigentlich einer eigenen Abhandlung, doch kurz zusammengefasst: Die Korrelation ist mit einer anderen Operation eng verwandt, die Faltung oder Konvolution heißt. Der Unterschied besteht darin, dass beim Berechnen das Signalmuster in der Zeit umgekehrt und folglich von hinten nach vorn durchlaufen wird. Für die Faltung ist im Frequenzbereich ein schneller Algorithmus möglich. Holen Sie jetzt einfach tief Luft und sprechen Sie mir nach:

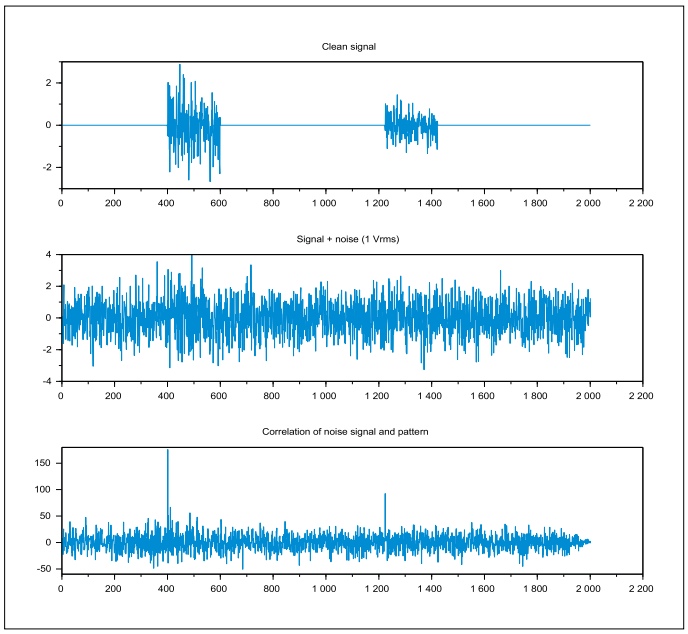



Bild 6. Komplexe Signalmuster, hier ein Pseudo-Random-Signal, liefern die besten Ergebnisse. In der Korrelation erscheinen schmale Impulse, denn Übereinstimmung ist nur zu einem einzigen Zeitpunkt vorhanden.

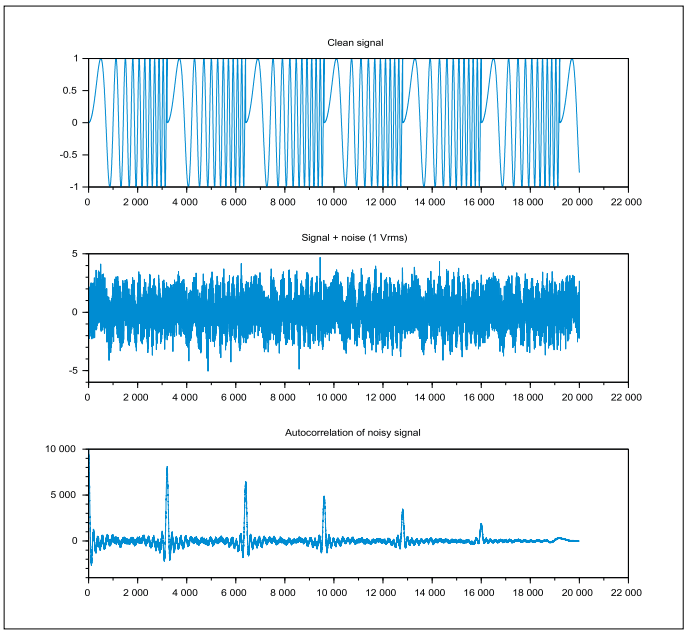



Bild 7. Mit der Autokorrelation lässt sich die Periode eines Signals bestimmen.

„Die FFT einer Faltung ist die Term-für-Term-Multiplikation der FFTs zweier Signale“, dabei treten konjugiert komplexe Größen auf. Diese Methode reduziert die Anzahl der Operationen drastisch, falls die Signale lang sind. Präziser formuliert: Die Anzahl der Operationen wird von N2 auf 3 · N · log(N) reduziert, der Unterschied steigt mit wachsendem N. Mehr Lektüre zu diesem Thema ist für unerschrockene Leser unter den Quellenangaben [3]...[7] vermerkt.

## Autokorrelation

Während die Korrelation das Auftreten eines Signalmusters in einem anderen Signal betrifft, bezieht sich die Autokorrelation auf die Ähnlichkeit eines Signals mit sich selbst. Die Autokorrelation verwendet den gleichen Algorithmus, das Signal ist jedoch gleichzeitig das Signalmuster. Was lässt sich damit anfangen? Da das Signal nach einer zeitverschobenen Kopie von sich selbst abgetastet wird, lässt sich der Zeitpunkt bestimmen, an dem die Wiederholung des Signals beginnt. Die Zeit zwischen solchen Zeitpunkten ist die Periode dieses Signals. In Bild 7 ist ein so genanntes gewobbeltes Signal dargestellt. Es besteht aus einer Reihe von Sinusschwingungen mit linear ansteigender Frequenz. Radar-Spezialisten nennen ein Segment dieses Signals einen Chirp, weil es sich so anhört, wenn das Signal einen Lautsprecher steuert. Mit der Autokorrelation ist es trotz des Rauschens möglich, die Wiederholungszeit der Chirps zu erkennen, also die Periode des Signals zu bestimmen.

## Schlussbetrachtung

Die Effizienz dieser wenig aufwendigen Methode ist verblüffend hoch. Wenn Sie meinen Beitrag über die digitalen FIR-Filter [9] noch einmal lesen, werden Ihnen dort Inhalte begegnen, die mit dem hier behandelten Thema in engem Zusammenhang stehen. Ich möchte Sie ermutigen, den Stoff durch Herunterladen von Scilab [2] und Nachvollziehen der Beispiele zu vertiefen.

(160374)gd

Dieser Beitrag ist auch in Circuit Cellar Nr. 299 erschienen.

## Quellen und Weblinks

[1] Korrelation (Bedeutung): www.dwds.de/wb/Korrelation

[2] Scilab (Numerisches Rechen-Tool, gratis): www.scilab.org

[3] Numerical recipes in C: The art of scientific computing, William H. Press & all, Chapter 13.2 Correlation and auto-correlation using FFT, ISBN 0-521-43108-5

[4] Understanding Digital Signal Processing (3rd Edition), Richard G. Lyons, ISBN 978-0137027415

[5] Signal processing and correlation techniques, Peter T. Gough, University of Canterbury, Christchurch, New Zealand: https://www.scribd.com/document/149011617/UIA-Correlation

[6] Digital Signal Processing: A Computer Science Perspective, Chapter 9 Correlation, Jonathan Y. Stein, Editions John Wiley & Sons, Inc., ISBN 0-471-29546-9

[7] The Scientist and Engineer’s Guide to Digital Signal Processing, Steven W. Smith: www.dspguide.com/ch7/3.htm

[8] Wikipedia, Kreuzkorrelation: https://de.wikipedia.org/wiki/Kreuzkorrelation

[9] Digitale Filter einfach erklärt: FIR-Filter, Elektor Dezember 2016: www.elektormagazine.de/150635

# Impedanz-Analyzer

Von **Wil Dijkman (NL)**

Mit einem LCR-Meter sind die Impedanzen und Verluste von Kondensatoren und Induktivitäten bei einer oder mehreren Frequenzen messbar. Wenn diese Komponenten in einem breitbandigen Übertragungsweg liegen, ist jedoch der Verlauf über einen weiten Frequenzbereich von Interesse. Hier hat der Impedanz-Analyser sein Einsatzfeld, denn er stellt Impedanzen und Verluste über der Frequenz anschaulich als Grafik dar.

**Kurzer Exkurs in die Geschichte**

Impedanz-Analyser sind komplexe Messgeräte, die in kleinen Betrieben oder im Heimlabor eher selten anzutreffen sind. Die Entwicklung der heute bekannten Geräte geht auf die Dekade 1960 bis 1970 zurück. Vor jener Zeit gab es bereits LCR-Meter, die meistens nach dem Prinzip der Messbrücke arbeiteten. Erst als die Mikrocontroller zum Allgemeingut wurden, konnten komplexe Messgeräte wie der Impedanz-Analyser gebaut werden. Einer der ersten und vermutlich bekanntesten Typen ist das Modell 4194 von Hewlett-Packard, siehe **Bild 1**. Die Handicaps dieses monströsen Geräts, das bis heute in vielen Entwicklungslabors zum Inventar gehört, sind seine Größe, sein Preis, sein Energiebedarf sowie die ungewöhnlich vielen Bedienelemente. Auf der anderen Seite ist der Hewlett-Packard 4194 ein solides und zuverlässiges Messgerät, das seine

# Eher seltenes, aber nützliches Messgerät

Aufgaben voll und ganz erfüllt. Nur wenige andere Messgeräte-Modelle dürften 40 Jahre Einsatzpräsenz überdauert haben. Während der genannten Dekade wandelten viele andere Messgeräte ihr Gesicht und ihr Innenleben: Gemeint sind Multimeter, Oszilloskope, Signalgeneratoren, Spektrum-Analyser und ähnliches mehr. Der Impedanz-Analyser trotzte diesem Trend, er blieb einfach das, was er von Anfang an war.

Nebenbei bemerkt: Für einen längeren Exkurs in die Geschichte empfiehlt der Autor das Studium einer Arbeit von Henry P. Hall im Internet [1].

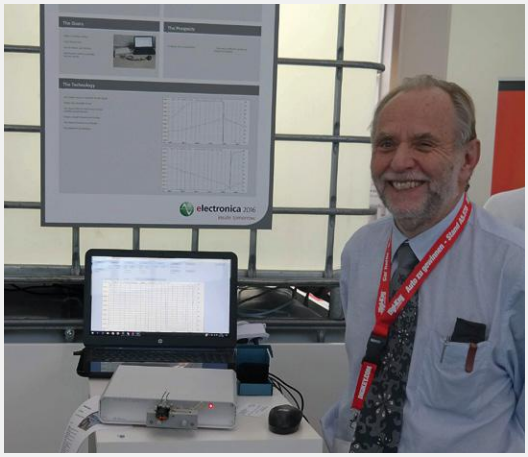
**Impedanz-Analyser, wie geht das?**

Der Autor hat viele nützliche Informationen dem *Impedance Measurement Handbook* entnehmen können, das gegenwärtig bei *Keysight Technologies* im Internet steht [2].

Das Prinzip der Impedanzmessung ist in **Bild 2** skizziert. Die zu messende Impe-

## Fast Forward Award 2016

Der Impedanz-Analyser gehörte zu den Projekten, die sich um den „Elektor Fast Forward Award 2016“bewarben. Dieser Wettbewerb wurde von Elektor in Zusammenarbeit mit der Fachmesse „electronica“ in München veranstaltet.
https://youtu.be/AqmPsjSF00Y



danz wird dort *Device Under Test* (DUT) genannt. Infolge der hohen Opamp-Verstärkung ist die Spannung am invertierenden Eingang nahezu Null. Da die Eingangsimpedanz des Opamp hoch ist, fließt der Strom, der durch das Messobjekt DUT fließt, auch durch $R_{ref}$. Daraus lässt sich herleiten:

$$Z_{DUT} = \frac{-V_x}{V_r} \times R_{ref}$$

Falls das Signal eine Gleichspannung ist, lässt sich die Impedanz unkompliziert messen. Wenn es sich um eine Wechselspannung handelt, sind phasenempfindliche Messmittel nötig, um den Realteil und Imaginärteil von $V_x$ und $V_r$ korrekt zu ermitteln. Die Impedanz des Messobjekts ergibt sich durch die Addition der komplexen Anteile von $V_x$ und $V_r$. Mit diesem Wissen im Hintergrund müssen folgende Aufgaben gelöst werden:

- Die Messanordnung mit dem Opamp ist im Prinzip gut brauchbar, sie lässt sich jedoch nur für Frequenzen bis zu einigen 10 kHz oder vielleicht 100 kHz realisieren. Darüber hinaus machen Größen wie Phasenverschiebung, Slew Rate, Propagation Delay Time oder das Absinken der Eingangsimpedanz die Methode zunichte. Dieses Problem bedarf einer Lösung.
- Das Eingangssignal muss von einem steuerbaren Signalgenerator erzeugt werden.
- Das Messsignal muss zunächst verstärkt und dann phasenabhängig bewertet werden. In diesen Ablauf wird der steuerbare Signalgenerator einbezogen.
- Die Messergebnisse sind grafisch als Diagramme darzustellen. Die Software steuert auch die Messfrequenz und die Amplitude des Eingangssignals, weil sich die Impedanz möglicherweise nicht linear verhält. Ferner kann die Software systematische Messfehler korrigieren.

Bild 1. Der Hewlett-Packard HP4194A wiegt 37 kg, er hat einen Leistungsbedarf von 400 VA. Die Betriebsanleitung umfasst 33 Seiten, auf weiteren 3 Seiten stehen die Technischen Daten (Foto: SGLabs).

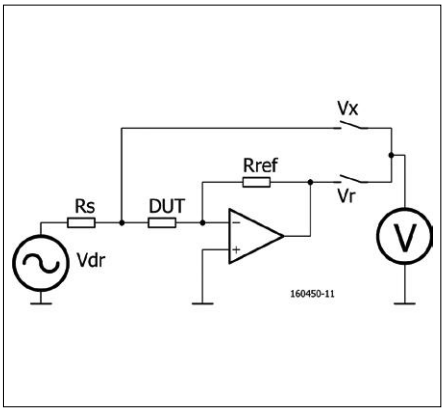



Bild 2. Eine Methode, mit der Impedanzen gemessen werden können.

### Automatische Messbrücke

Ein Opamp, der auch bei hohen Frequenzen mit genügend Verstärkung (etwa 200), minimaler Phasenverschiebung und anderen hier erwünschten Eigenschaften arbeitet, ist nicht leicht zu finden. Da es sich um ein rückgekoppeltes System handelt, das auch bei Frequenzen um 40 MHz stabil bleiben muss, darf die Phasenverschiebung höchstens ein Viertel der Periodendauer 25 ns betragen, das sind 6,5 ns. Hinzu kommt, dass die Gegenkopplung variabel sein muss, damit sich mehrere Messbereiche realisieren lassen. Damit ergeben sich zusätzliche Hürden, sie entstehen, weil das System seine Stabilität im gesamten Frequenzbereich bewahren muss.
Der Autor hat deshalb nach einer anderen Lösung gesucht. Wegweisend war die Überlegung, dass ein breitbandiger Verstärker nicht zwingend notwendig ist. Bei jeder Einzelmessung wird das Eingangssignal vom System bei einer definierten Frequenz gemessen, so dass ein schmalbandiger Verstärker genügt. Es muss lediglich dafür gesorgt werden, dass die Frequenz, auf die der Verstärker abgestimmt ist, mit der Frequenz des Eingangssignals mitläuft. Diese Methode ist in **Bild 3** skizziert.

Das System besteht aus mehreren Phasendetektoren, zwei Opamps, die als Integratoren arbeiten, sowie zwei Modulatoren. Dieser Konstellation sind Pufferstufen vor- und nachgeschaltet. Angenommen die Frequenz des Eingangssignals stimmt mit der Frequenz des Signals überein, das die Phasendetektoren und Modulatoren steuert. Die Steuersignale des Phasendetektors und des Modulators im oberen Zweig sind bezogen auf den unteren Zweig um 90° phasenverschoben. Das Signal $V_{in}$ soll ein beliebiges sinusförmiges Signal sein:

$V_{in} = a \sin(\omega t) + b \cos(\omega t)$

Durch Demodulieren von $V_{in}$ mit den Steuersignalen entsteht:

$V_1 = (a \sin(\omega t) + b \cos(\omega t)) \sin(\omega t) = (a - a \cos(2\omega t) + b \sin(2\omega t))/2$
$V_2 = (a \sin(\omega t) + b \cos(\omega t)) \cos(\omega t) = (b + b \cos(2\omega t) + a \sin(2\omega t))/2$

Beim Integrieren von $V_1$ und $V_2$ fallen die Sinus- und Cosinus-Ausdrücke heraus. Außerdem fügt ein Integrator-Opamp die Verstärkung –A mit folgendem Ergebnis hinzu:

$V_3 = -(A/2)\ a$
$V_4 = -(A/2)\ b$

Das Modulieren mit den Steuersignalen ergibt:

$V_5 = -(A/2)\ a \sin(\omega t)$
$V_6 = -(A/2)\ b \cos(\omega t)$

Schließlich werden diese beiden Signale miteinander addiert:

$V_7 = -(A/2)\ (a \sin(\omega t) + b \cos(\omega t)) = -(A/2)\ V_{in}$

Damit ist das Ausgangssignal ein verstärktes und invertiertes Abbild des Eingangssignals.

Die Methode ist in der Praxis für Messsignale mit Frequenzen bis zu etlichen 10 MHz geeignet. Die Bandbreite des zu konstruierenden Bandpasses wird von den Integratoren bestimmt, die Signalverstärkung beginnt fast bei 0 Hz. **Bild 4** und **Bild 5** geben die ausgearbeitete Schaltung wieder.
In Bild 4 passiert das Signal die Eingangspuffer T5 und T4, von dort gelangt es zu den wechselstromgekoppelten Demodulatoren. Dargestellt ist nur der Zweig für das SIN-Signal, der andere Zweig ist identisch. Die klassisch geschalteten Demodulatoren können hohe Signalfrequenzen verarbeiten. Es handelt sich um zwei sogenannte *Long-tailed Pairs* (T2A/T2B und T3A/T3B), wobei die Basen und die Kollektoren jeweils miteinander verbunden sind. Das Eingangssignal wird den Basen zugeführt, während das Steuersignal an den Emittern liegt. Ein weiteres *Long-tailed Pair* (T1A/T1B) setzt die steuernde Spannung in einen Strom um. Vom Modulator gelangt das Signal symmetrisch zum Integrator IC1, der Ausgang von IC1 führt zum Modulator in Bild 5.
Der Aufbau des Modulators ist mit dem Demodulator nahezu identisch. Der Unterschied besteht darin, dass das Eingangssignal des Demodulators sinusförmig ist und das Ausgangssignal die gesuchte Gleichspannungskomponente enthält. Beim Modulator ist das Eingangssignal eine Gleichspannung, während das Aus-

## Eigenschaften

- Messgrößen: |Z|, Φ, L, C, D, Q, R, X
- 8 Kombinationen sind bei der Darstellung möglich
- Messfrequenz: 100 Hz...40 MHz, Auflösung 42 mHz
- Niveau des Messsignals (Amplitude): 40 mV...1 V in 4 Schritten
- Maximaler Messstrom (Amplitude): 20 mA
- Keine Gleichstromkomponente
- Grundgenauigkeit besser als 1 %

| Parameter | Bereich | Auflösung |
|---|---|---|
| Z, R, X | 100 mΩ...10 MΩ | 3 mΩ |
| Φ | ±90 ° | 0,01 ° |
| L | 10 nH...10 kH | |
| C | 100 fF...0,01 F | |
| D | 0,001...10 | |
| Q | 0,1...1000 | |

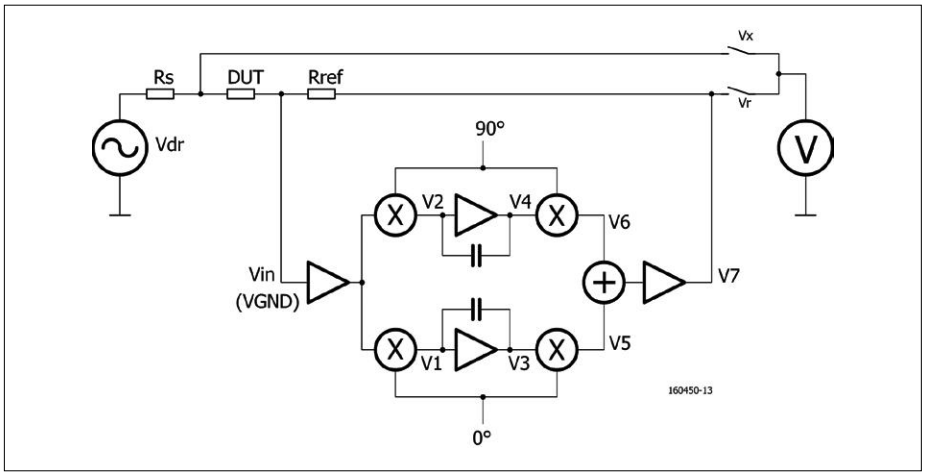



Bild 3. Automatische Messbrücke zum Messen von Signalen hoher Frequenzen.

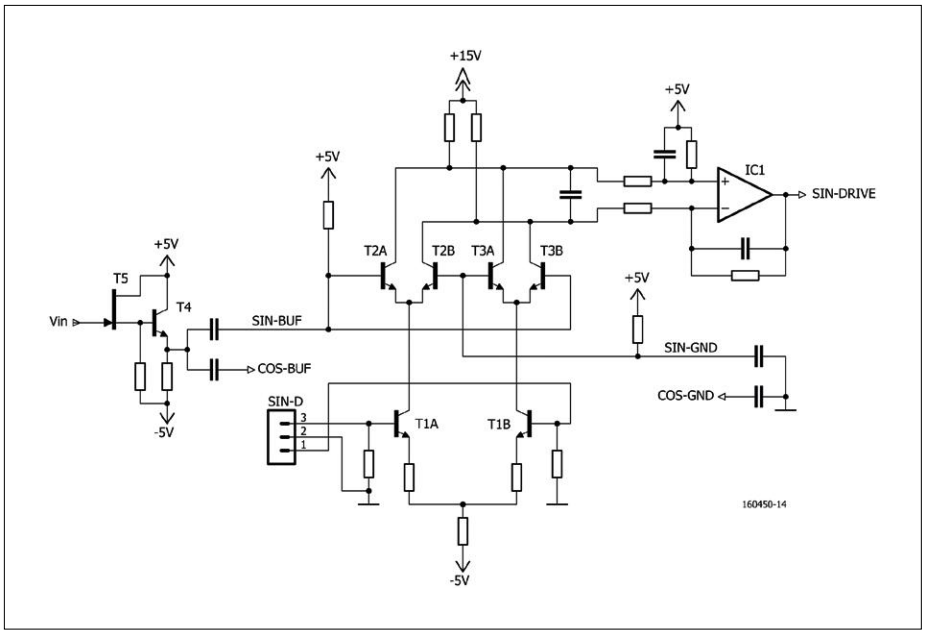



Bild 4. Die Eingangsstufe (T4/T5), der Demodulator und der Integrator (IC1). Nur der Sinus-Kanal ist dargestellt, der Cosinus-Kanal (hinter T4/T5) ist identisch.

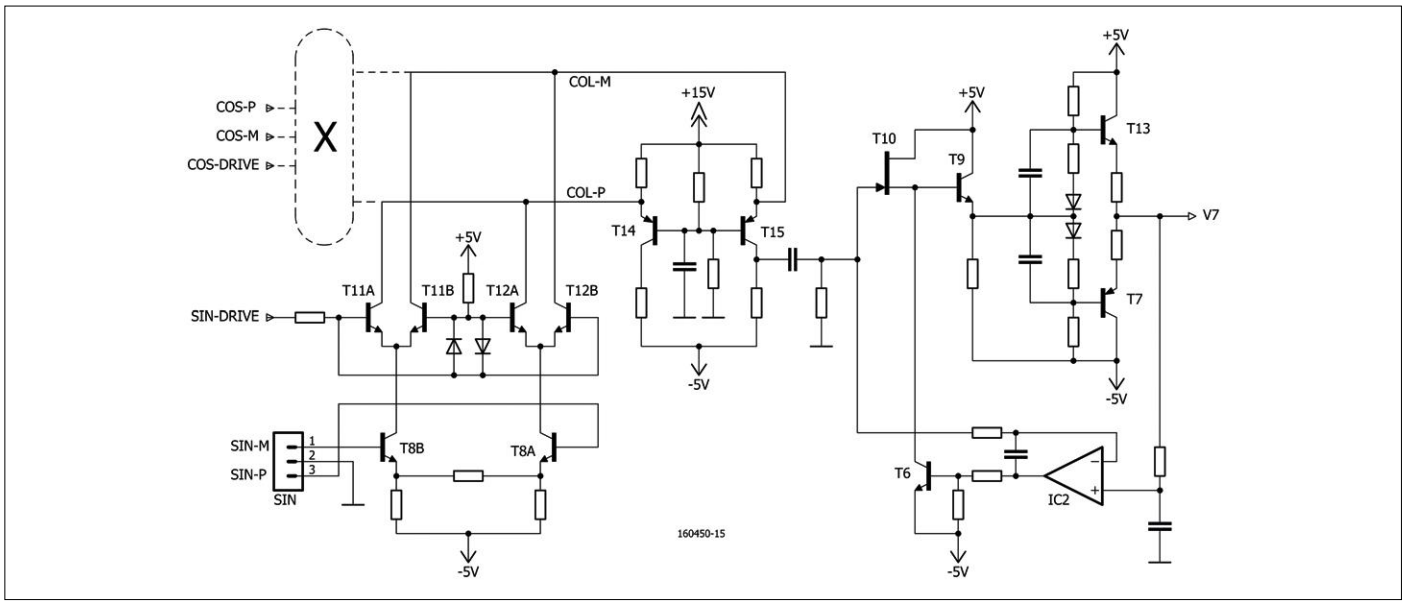



Bild 5. Ein Modulator, der Addierer (T14/T15) und der Endverstärker (rechts). Der zweite Modulator ist nur symbolisch (gestrichelt) dargestellt.

gangssignal Sinusform hat. Das symmetrische Ausgangssignal des Modulators, COL-P und COL-M, wird zusammen mit dem Ausgangssignal des COS-Zweiges dem Addierer mit T14 und T15 zugeführt. Dabei ist zu beachten, dass es sich um Ströme handelt, so dass die Ausgänge COS und SIN einfach miteinander zusammengeschaltet werden können.

Das Signal des Addierers wird asymmetrisch ausgekoppelt, über einen Kondensator gelangt es zum Ausgangspuffer. Er besteht aus dem Source- und dem Emitterfolger T10 und T9, gefolgt von einer einfachen Klasse-AB-Endstufe. Der Offset des Ausgangs wird von IC2 stabilisiert. Dieser Opamp vergleicht das Eingangssignal mit dem Signal, das am Gate von T10 liegt und stellt den Strom dieses Transistors so ein, dass die Differenz gegen Null geht. Die Verstärkung der Endstufe beträgt etwas weniger als 1. Die Endstufe kann mit 20 mA belastet werden, hier bildet der Referenzwiderstand die Last. Die Verstärkung der Schaltung liegt bei ungefähr 200.

## Messverstärker und Schalter

Das Konzept des Impedanz-Analysers geht aus **Bild 6** hervor. Links außerhalb des Bildes befindet sich die Signalquelle $V_{dr}$, von hier fließt ein Strom über $R_S$ durch das Messobjekt DUT und $R_{ref}$. Der Verbindungspunkt von $R_{ref}$ und dem Messobjekt ist mit dem Eingang der automatischen Messbrücke verbunden. Dieser Schaltungspunkt wird von der Messbrücke soweit möglich auf Massepotential gehalten (virtuelle Masse, *Virtual Ground*, VGND). Der Ausgang der automatischen Messbrücke liegt an $R_{ref}$. Um mehrere Messbereiche zu realisieren, ist $R_{ref}$ zwischen 50 Ω und 100 kΩ über Relais umschaltbar. Infolge der Relais liegt eine niedrige zusätzliche Impedanz mit $R_{ref}$ in Reihe, so dass ein geringer Messfehler entsteht. Die zusätzliche Impedanz beträgt etwa 20 nH, bei 40 MHz entspricht

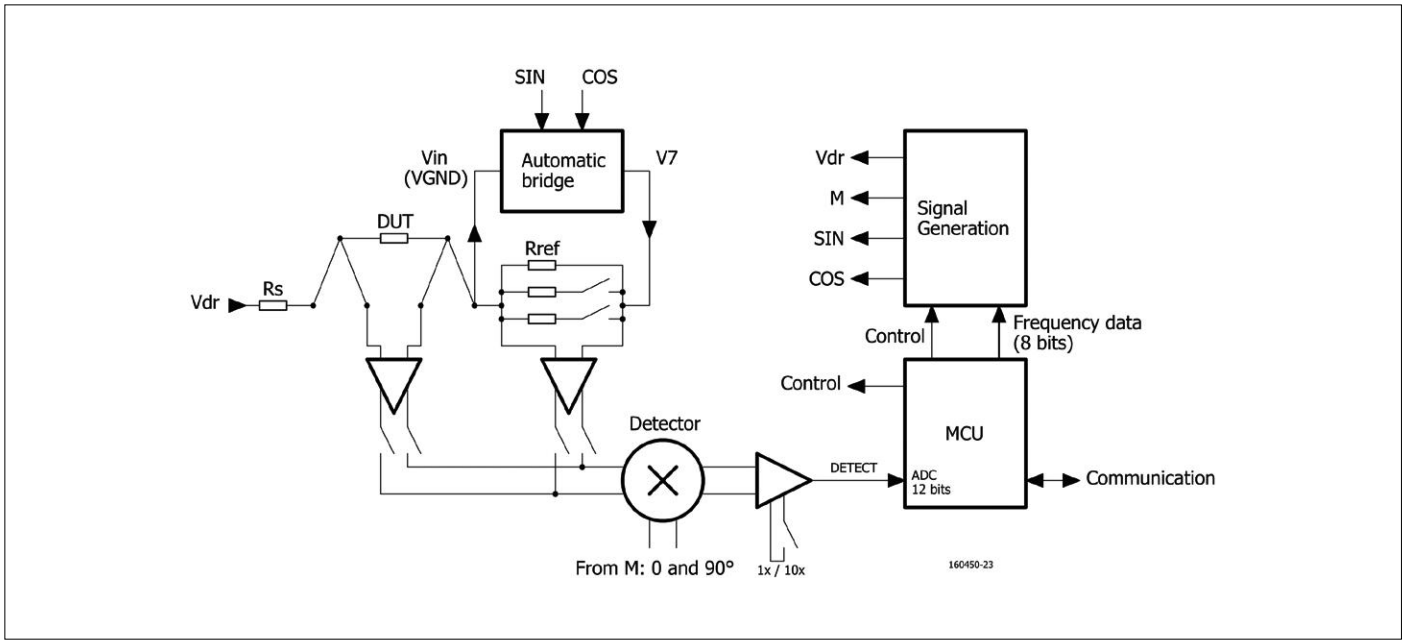



Bild 6. Blockschematische Darstellung des Impedanz-Analysers.

dies ungefähr 5 Ω. Der Messfehler wird später von der Software kompensiert.
Der Strom durch die Leitungen zum Messobjekt kann Spannungsabfälle verursachen, die nicht in die Messung eingehen sollen. Deshalb ist das Messobjekt mit dem Messverstärker über eigene Leitungen verbunden, über die nur wenig oder kein Strom fließt. Eine Messung mit dieser Art der Leitungsführung wird auch „Vierpunktmessung" genannt. Die Messsignale werden über einen Differenzverstärker am Messobjekt oder am Referenzwiderstand abgegriffen. Der Differenzverstärker muss mit hoher Bandbreite und hoher Stabilität arbeiten. Am Verstärkerausgang liegt ein Schalter, der in geschlossenem Zustand keine Signal- oder Bandbreite-Verluste verursachen darf, in offenem Zustand muss das Signal möglichst wirksam unterdrückt werden.
Die Schaltung der beiden Differenzverstärker ist aus **Bild 7** ersichtlich. Da die Eingangsimpedanz hoch sein muss, liegen JFETs an den Eingängen. Allerdings hat dies den Nachteil, dass die Gleichspannungseinstellung nicht genau definiert ist. Deshalb wird die Einstellung durch Variieren des JFET-Stroms stabilisiert, so dass sich die Gate-Source-Spannung ändert. Der Opamp stellt die Spannung an den Emittern der *Long-tailed-pair*-Transistoren auf den gleichen Wert wie die Eingangsspannung ein. Diese Maßnahme entspricht der Methode, die beim Signal $V_{DRIVE}$ im Ausgangsverstärker angewendet wird. Der Preis, der dafür gezahlt werden muss, ist eine etwas niedrigere Verstärkung bei den hohen Frequenzen im Vergleich zu den niedrigen Frequenzen. Vom Eingang bis zum zugehörigen Emitter ist die Verstärkung bei den niedrigen Frequenzen gleich 1. Bei den hohen Frequenzen hängt die Verstärkung von den Eigenschaften des Sourcefolgers und des anschließenden Emitterfolgers ab.
Das Prinzip der Schalter zeigt **Bild 8**. Diese Schaltung bezieht sich auf eine Hälfte eines Differenzverstärkers, für jeden Differenzverstärker ist sie doppelt ausgeführt. Insgesamt sind zwei Differenzverstärker vorhanden.
Schalterfunktion haben die Dioden D1, D2, und D3, so genannte *Bandswitching*-Dioden, sie haben folgende wichtige Eigenschaften:

- niedriger Durchlasswiderstand im Leitzustand,
- niedrige Kapazität im sperrenden Zustand.

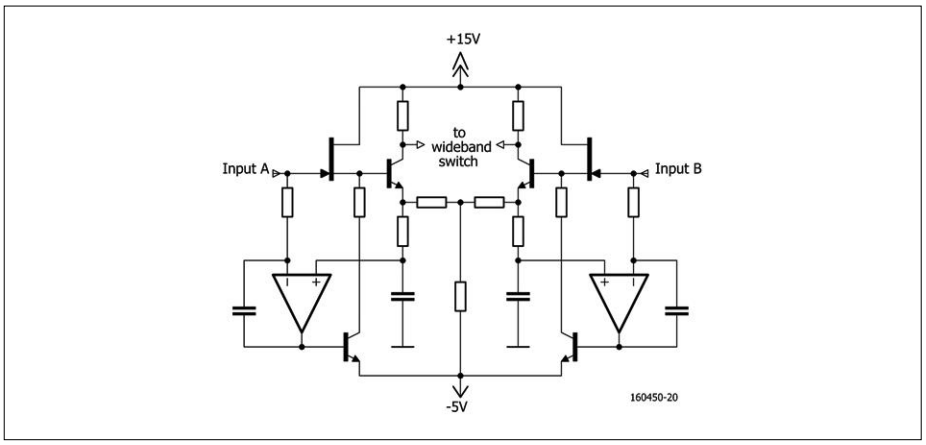



Bild 7. Die Gleichspannungseinstellung der JFET-Eingänge wird bei den Differenzverstärkern durch Opamps stabilisiert.

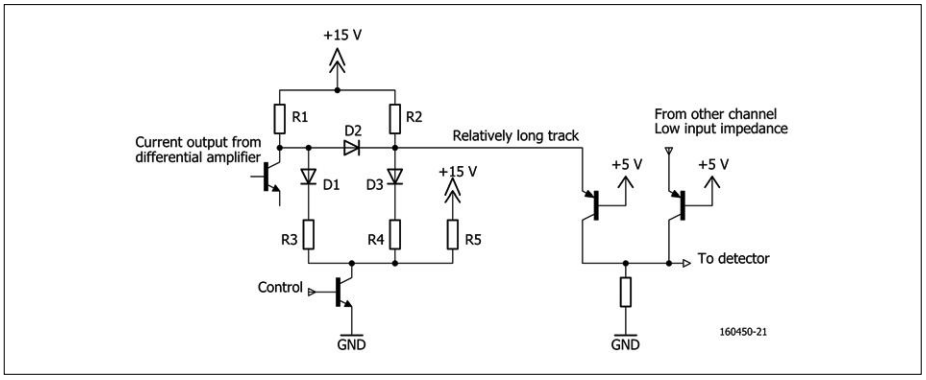



Bild 8. Prinzip der im Messsystem verwendeten Schalter.

Sperrende pn-Übergänge können die Signale nicht so stark unterdrücken, dass ein Übersprechen zwischen den Messverstärker-Signalen unmöglich ist. Deshalb wird hier doppelt geschaltet: Wenn das steuernde Signal niedrig ist, sperren D1 und D3, denn über R5 werden die Kathoden nach +15 V gezogen. In diesem Fall leitet D2. Der durch R2 fließende Strom gelangt teilweise zum Kollektorausgang des Differenzverstärkers. Der Signalstrom des Differenzverstärkers fließt jetzt über D2 zum Emitter des PNP-Transistors. Dieser Schaltungsteil hat eine vergleichsweise niedrige Impedanz. Auch wenn das Signal über eine

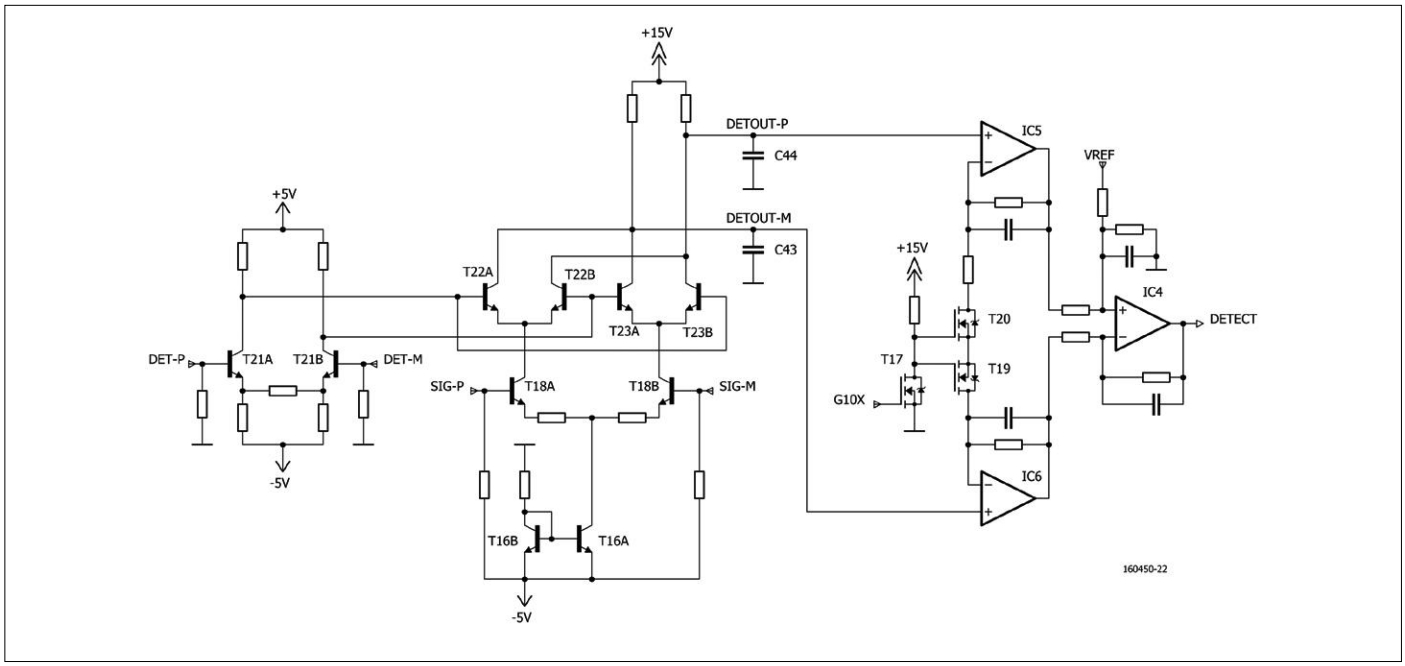



Bild 9. Detektor mit nachgeschaltetem Gleichspannungsverstärker.

lange Leiterbahn auf der Platine transportiert werden muss, entstehen bei den hohen Signalfrequenzen nur wenig Verluste, denn die Signalspannung ist niedrig. Der Signalstrom wird von der Kapazität einer langen Leiterbahn kaum beeinflusst. Der Kollektorwiderstand des PNP-Transistors setzt den Signalstrom wieder in eine Spannung um.

Wenn das steuernde Signal hoch ist, leiten D1 und D3. R2 und R4 werden so gewählt, dass der PNP-Transistor sperrt und die Werte von R1 und R3 sind so bemessen, dass auch D2 sperrt. Zwei sperrende Übergänge bedeuten in der Praxis, dass das Signal genügend gedämpft wird.

## Der Detektor

Der Detektor in **Bild 9** ist mit den zuvor betrachteten Modulatoren und Detektoren fast identisch. Vorhanden sind zwei Eingangssignale: Steuersignal M, in der Schaltung DET-P und DET-M, sowie das Ausgangssignal des Messverstärkers, SIG-P und SIG-M. Die Ausgangssignale DETOUT-P und DETOUT-M werden von C43 und C44 geglättet. Danach folgt ein Gleichspannungs-Messverstärker, der das Signal wahlweise mit dem Faktor 1 oder 10 verstärkt und ein asymmetrisches Signal liefert. Dieses Signal wird dem Analog/Digital-Wandler (ADC) des Mikrocontrollers zugeführt.

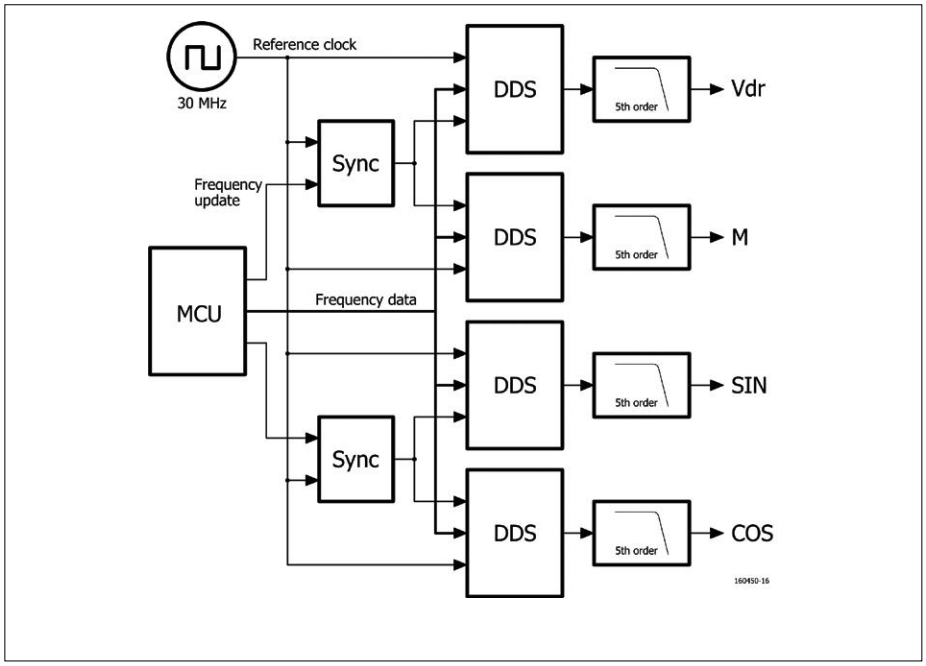



Bild 10. Vier integrierte Direct Digital Synthesizer (DDS) generieren die Mess- und (De-) Modulationssignale.

## Die Signalgeneratoren

Was die steuernden Signale betrifft, sind insgesamt vier Signale nötig. Das erste Signal ist das Eingangssignal für das Messobjekt DUT. Die Amplitude muss variabel sein, damit nichtlineare Impedanzen kompensiert werden können. Dieses Signal wird mit $V_{dr}$ bezeichnet. Die nächsten beiden Signale sind die Steuersignale für die Messbrücke. Sie müssen gegeneinander um exakt 90° über den gesamten Frequenzbereich phasenverschoben sein. Dies sind die Signale SIN und COS. Ferner wird ein Steuersignal für den Phasendetektor benötigt. Bezogen auf $V_{dr}$ muss dieses Signal entweder in Phase oder um exakt 90° phasenverschoben sein. Dieses Signal ist mit M (von „Messen“) bezeichnet. Sämtliche Signale müssen die gleiche Frequenz haben, die Frequenz muss von 100 Hz bis 40 MHz variierbar sein.

Im Impedanz-Analyser 4194 von

Hewlett-Packard werden die Signale von komplexen PLL-Schaltungen generiert. In diesem Projekt kommt der *Direct Digital Synthesizer* (DDS) AD9851 von Analog Devices [3] zum Einsatz. Blockschematisch sind die Signalgeneratoren in **Bild 10** dargestellt. Der AD9851 kann Signale mit beliebigen Frequenzen im Bereich 0...70 MHz mit einer Auflösung von 0,042 Hz (180 MHz / $2^{32}$) generieren, hier wird jedoch nur der Bereich 100 Hz...40 MHz genutzt.

Den Ausgängen der DDS-Bausteine ist jeweils ein Tiefpass 5. Ordnung nachgeschaltet, der die Spiegelsignale und andere Mischprodukte ausfiltert. Die Phasendifferenz zwischen den Signalen SIN und COS beträgt konstant 90°, während die Phasenverschiebung zwischen den Signalen $V_{dr}$ und M abhängig vom Messzyklus abwechselnd 0° und 90° beträgt. Die Phasenverschiebung zwischen $V_{dr}$ und M einerseits und SIN und COS andererseits ist nicht definiert, nur die Frequenz aller Signale stimmt überein.

Die Amplitude des Signals $V_{dr}$ muss variierbar sein. Dies wird über den $R_{set}$-Eingang des AD9851 realisiert, in **Bild 11** ist das Prinzip skizziert. Die drei MOSFETs können $R_{set}$ in vier Stufen anpassen (eigentlich acht, doch nur vier sind sinnvoll) und damit die Amplitude des DDS-Ausgangssignals einstellen.

Das gefilterte, symmetrische Ausgangssignal steuert einen wechselspannungsgekoppelten Verstärker, er formt daraus das asymmetrische Signal für den Strom, der durch das Messobjekt DUT fließt. Der Verstärker ist eine einfache AB-Endstufe, er ist mit dem DUT unmittelbar verbunden. Der Nullpunkt des Ausgangs wird mit einem Opamp stabilisiert. Der Opamp vergleicht den Eingang der Endstufe mit dem Ausgang, er korrigiert die Differenz, indem er den Strom durch den JFET erhöht oder verringert, was sich auf die Gate-Source-Spannung auswirkt. Dieser Verstärker ist mit dem Ausgangsverstärker der Messbrücke aus Bild 5 identisch.

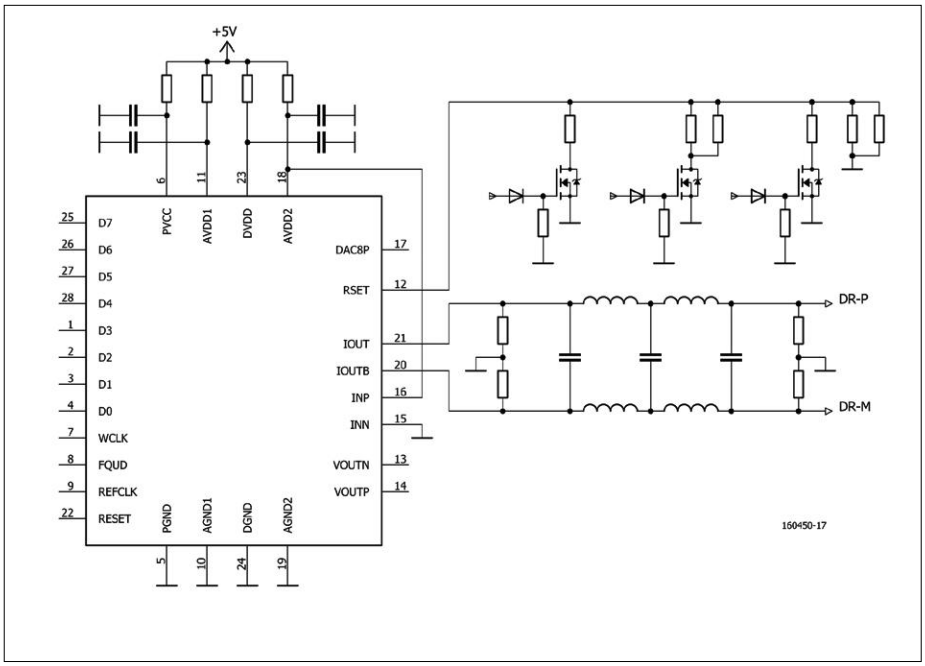



Bild 11. Prinzip eines DDS-Signalgenerators mit Tiefpass 5. Ordnung. Die MOSFETs sind Teil der Amplitudeneinstellung beim Signal $V_{dr}$.

## Die Software des Mikrocontrollers

Die „Intelligenz" des Impedanz-Analysers steckt in einem Mikrocontroller PIC18F4523 von Microchip. An den Mikrocontroller werden keine speziellen Anforderungen gestellt, er muss lediglich unkomplizierte Steuerungsaufgaben übernehmen. Zu seiner Ausstattung müssen ein serieller Port, ein 12-bit-AD-Wandler sowie eine genügende Anzahl Ein- und Ausgänge gehören. Die Software für den Mikrocontroller wurde in mikroBasic von MikroElektronika geschrieben. Wenn der Impedanz-Analyser vom Host-Rechner einen Messauftrag erhält, mit der Messfrequenz und der Amplitude des Eingangssignals als Parameter, arbeitet er folgende Routine ab:

- Einstellen der Frequenz.
- Einstellen der Signalamplitude.
- Messempfindlichkeit auf Maximum (Verstärkung=10).
- (1) Messen der Signalamplitude am Messobjekt (DUT), sowohl in Phase als auch 90° phasenverschoben bezogen auf das Signal $V_{DRIVE}$.
- Bei Übersteuerung des Messverstärkers: Verstärkung auf 1, dann zurück nach Punkt (1).
- (2) Verstärkung wieder auf 10 und Messungen der Spannung an Messwiderstand $R_{ref}$ wiederholen.
- Bei Übersteuerung des AD-Wandlers zurück nach Punkt (2).
- Mit den gefundenen Messwerten wird $Z_{DUT}$ berechnet (komplexe Zahlen).
- Falls (der Modulo des berechneten Werts von) $Z_{DUT}$ viel größer oder viel kleiner als $R_{ref}$ ist, muss $R_{ref}$ angepasst werden: Verstärkung wieder

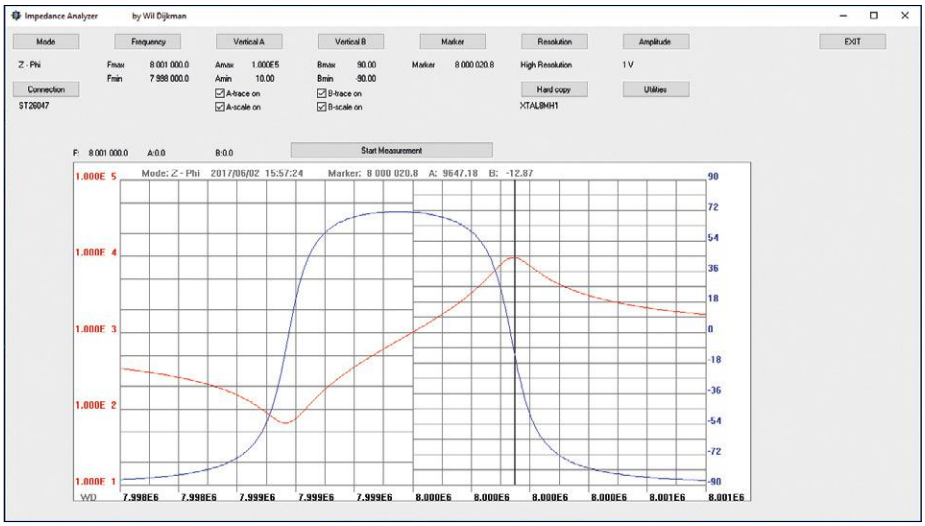



Bild 12. Die Steuerung und die Benutzeroberfläche laufen auf einem PC. Die Grafik kann zwei Parameter gleichzeitig darstellen.

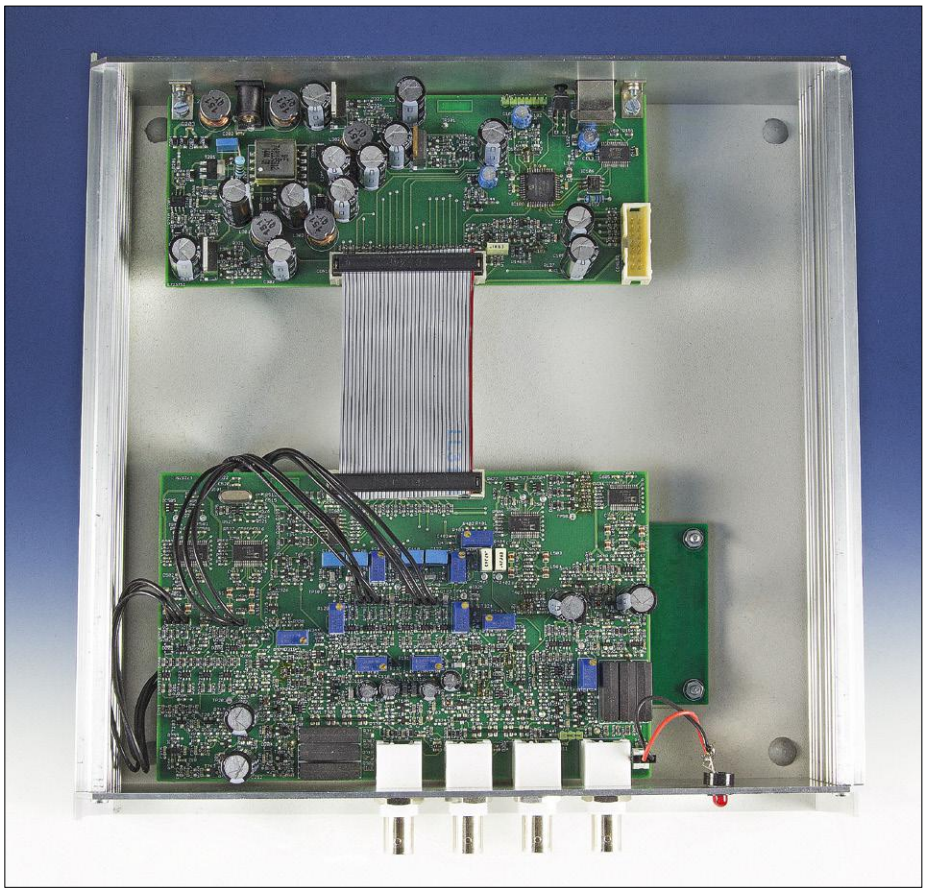


Bild 13. Blick in das Innere des Impedanz-Analysers, eingebaut in ein Gehäuse mit den Abmessungen 25 · 25 · 5 cm.

auf 10, zurück nach Punkt 1.

- Falls der gefundene Wert von $Z_{DUT}$ in gleicher Größenordnung wie $R_{ref}$ liegt, oder $R_{ref}$ steht auf dem maximalen oder minimalen Wert, ist die Messung abgeschlossen.
- Die gefundenen Werte R und X der komplexen Impedanz $Z_{DUT}$ werden dem Host übergeben.
- Warten auf den nächsten Messauftrag.

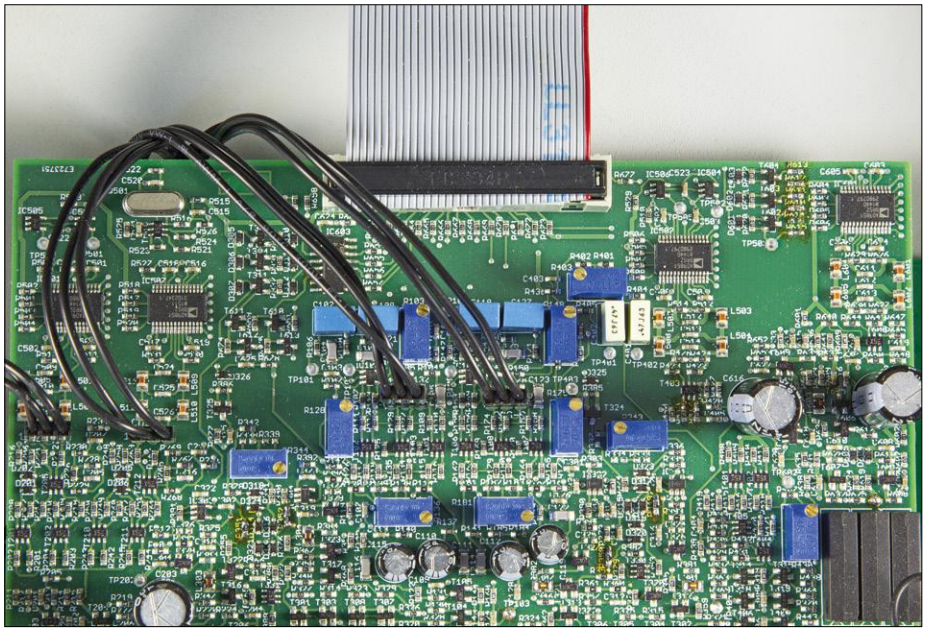


Bild 14. Die Hauptplatine aus der Nähe betrachtet. Gut erkennbar sind die vier DDS-ICs (oben) und die Relais zum Umschalten des Referenzwiderstands (rechts unten).

Der Referenzwiderstand $R_{ref}$ muss nicht bei jeder Messung angepasst werden, denn beim Durchlaufen des Frequenzbereichs liegen die meisten Messwerte dicht beieinander.

## Die Software für den Host-Rechner

Das Programm des Hosts muss unkompliziert bedienbar sein und ferner die systematischen Messfehler des Impedanz-Analysers korrigieren. **Bild 12** gibt einen Eindruck von der Benutzeroberfläche auf dem Bildschirm.

Nach dem Programmstart wird zuerst der COM-Port gesucht, an den die Messeinheit angeschlossen ist. Dann wird eine Messung mit den Einstellungen der vorangegangenen Sitzung durchgeführt.

In der Mitte über dem Grafik-Feld befindet sich die Schaltfläche „Start Measurement“. Wenn sie angeklickt wird, werden die Anfangs- und Endfrequenzen der letzten Messung übernommen. Dann berechnet das Programm die Werte, die zum Einstellen der Signalgeneratoren in der Messeinheit benötigt werden. Diese Berechnung übernimmt der PC, weil das Programm des Mikrocontrollers, geschrieben in mikroBasic, nur mit einem 4-Byte-Fließkomma-Algorithmus rechnen kann. Die niedrige Genauigkeit würde unzulässig hohes Rundungsrauschen zur Folge haben. Der PC sendet die errechneten Werte zusammen mit den Amplituden-Werten zur Messeinheit. Die vom PC empfangenen Messwerte werden in einer R-X-Tabelle gepuffert, dann werden daraus die Parameter der Grafik berechnet. Diese Art der Verarbeitung hat den Vorteil, dass die grafische Darstellung angepasst werden kann, ohne die Messung noch einmal wiederholen zu müssen.

Die Schaltfläche „Mode“ öffnet ein Menü, in dem der Darstellungsmodus der Messwerte einstellbar ist. Wählbar sind die Kombinationen Z/φ, $L_s/R_s$, $L_p/R_p$, L/Q, $C_s/R_s$, $C_p/R_p$, C/D und R/X. Der jeweils erste Parameter wird auf der vertikalen Achse, der zweite Parameter auf der horizontalen Achse dargestellt. Über die Schaltflächen „Vertical A“ und „Vertical B“ sind weitere Einstellungen der vertikalen Achse möglich.

Zum Anschließen des Messojekts DUT wird eine Messvorrichtung benutzt. Jede Messvorrichtung weist eine parasitäre Impedanz auf, ihr Einfluss muss korrigiert werden. Dazu wird für jeden Anschluss

die Impedanz im offenen und kurzgeschlossenen Zustand bei verteilten Frequenzen im gesamten Frequenzbereich gemessen. Durch Einbeziehen der gefundenen Werte in das Berechnen der am Messobjekt festgestellten Impedanz wird die „wahre“ Impedanz ermittelt (siehe auch [2], Anhang B). Über die Schaltfläche „Connection“ kann die Messvorrichtung ausgewählt werden.

Im Frequenz-Menü, das sich hinter der Schaltfläche „Frequency“ verbirgt, lassen sich die Grenzen des Frequenzbereichs einstellen. Die Skalierung ist logarithmisch oder linear möglich. Eine weitere Betriebsart ist „Spot“, der Impedanz-Analyser arbeitet dann bei einer einzelnen Frequenz $F_{max}$ als Messbrücke. Die Skalenteilung wird vom Programm an die Frequenzgrenzen und die Art der Skalierung angepasst.

Natürlich ist auch ein „Marker“ (Cursor) vorhanden. Er hat die Gestalt einer senkrechten Geraden, die präzise auf eine Frequenz gesetzt werden kann und auch ausblendbar ist. Wenn der Marker eingeblendet ist, kann die Position auch mit der Maus bestimmt werden. Auf dem Bildschirm ist im Grafik-Feld die Marker-Frequenz zusammen mit den am Messobjekt DUT gemessenen Werten neben der Frequenz ablesbar.

Die horizontale Auflösung lässt sich in vier Stufen einstellen. Bei niedriger Auflösung dauert die Messung kürzer als bei hoher Auflösung, es werden jedoch Pixel ausgelassen, so dass möglicherweise Details verloren gehen. Meistens führen „Medium Resolution“ oder „High Resolution“ zu den erwarteten Ergebnissen.

Die Amplitude des Eingangssignals ist über die Schaltfläche „Amplitude“ mit den Werten 40 mV, 120 mV, 340 mV und 1 V einstellbar. Vorteilhaft ist dies beim Messen nichtlinearer Impedanzen wie bei Halbleitern oder Induktivitäten mit Kern. Bei niedriger Amplitude können die Messwerte mehr Rauschen enthalten. Im Bereich niedriger Frequenzen wird die Amplitude begrenzt, damit die Detektor-Schaltung nicht übersteuert wird.

Übrig bleibt noch das Menü „Utilities“, unter dem sich diverse Submenüs befinden. Hier kann der Impedanz-Analyser kalibriert werden, und an der gewählten „Connection“ lässt sich die Impedanz im offenen und kurzgeschlossenen Zustand messen. Die Kalibrierungs- und Kompensationsdaten werden in mehreren Dateien auf dem Host gespeichert, weil im Speicher des Mikrocontrollers der Platz fehlt. Das bedeutet, dass die Kalibrier- und Kompensier-Prozedur beim Wechsel auf einen anderen Host noch einmal durchlaufen werden muss.

Es ist möglich, zwei oder mehr Messungen übereinander darzustellen, so dass geringe Unterschiede zwischen gleichartigen Komponenten entdeckt werden können. Die Messkurven identischer Messungen können auch fortlaufend übereinander geschrieben werden, so dass die Linienstärke zunimmt. Auf diese Weise entsteht ein Eindruck von der Wiederholbarkeit einer Messung. Aussagefähig kann dies sein, wenn an den Grenzen des Messbereichs gemessen werden muss.

Die Grafik kann ausgedruckt oder als Windows Bitmap (BMP) exportiert werden. Die Messwerte sind auch in einer CSV-Datei speicherbar. Dort sind sie durch Kommata separiert, so dass sie mit Programmen wie Excel verarbeitet werden können.

## Schlusswort des Autors

Ich hoffe, dass mein Beitrag Ihr Verständnis der analogen Elektronik bereichern konnte. Für mich war der Impedanz-Analyser ein ebenso spannendes wie lehrreiches Projekt. Auch wenn die Schaltungen unkompliziert erscheinen, war ich meistens erst mit der vierten oder fünften Version zufrieden. Ich denke, dass dieses Messgerät für kleine (oder auch größere) Entwicklungslabors oder auch für den Unterricht an berufsbildenden Schulen von Interesse sein kann. Mein Impedanz-Analyser fällt etwas handlicher und insbesondere kostengünstiger aus (vielleicht um den Faktor 20) als vergleichbare, von der Industrie gefertigte Modelle.

Fragen oder Anregungen, vorzugsweise in den Sprachen Niederländisch oder Englisch, richten Sie bitte an die Email-Adresse wil.dijkman@onsbrabant-net.nl. Ich würde mich auch freuen, wenn Sie meine Website [4] besuchen. ⏮

(160450)gd

### Weblinks

[1] Geschichte der Impedanzmessung: http://www.ietlabs.com/pdf/GenRad_History/A_History_of_Z_Measurement.pdf

[2] Keysight Technologies Impedance Measurement Handbook: http://literature.cdn.keysight.com/litweb/pdf/5950-3000.pdf

[3] AD9851: www.analog.com/AD9851

[4] Website des Autors: www.wildijkmanelectronics.nl

[5] Video: https://youtu.be/AqmPsjSF00Y

ELEKTOR ETHICS

# Das IoT: Verantwortung für alle Beteiligten

Von **Tessel Renzenbrink** (NL)

> ***„In der Rückschau war das rasante Wachstum des World Wide Web nur der Trigger, der jetzt die wahre Explosion auslöst, seit die ‚Dinge' das Netz erobern."***
>
> – Neil Gershenfeld, Begründer der FabLabs, 1999.

Der technische und gesellschaftliche Wandel, der den Einzug des Internets in den Alltag begleitete, verblasst gegen die Umwälzungen, die mit dem *Internet of Things* einhergehen. Das IoT lebt durch seine *Objekte*, sie sind mit hohen Rechenleistungen ausgestattet und über das Internet vielfältig vernetzt. Die Objekte können gegenseitig Informationen austauschen, autonom Entscheidungen treffen sowie Abläufe starten, ohne dass der Mensch beteiligt ist.
Mit dem IoT wächst das Internet parallel zu seinen klassischen Funktionen zum elektronischen Nervensystem heran. Gestützt auf das breite Spektrum der Sensoren und Aktoren sind jederzeit Eingriffe in die existierende reale Welt möglich. Wegen des unmittelbaren Eingreifens in das Geschehen können Störfälle zu unvorhersehbaren, folgenschweren Konsequenzen führen. Während bei Cyberangriffen in der Vergangenheit meistens „nur" Daten gestohlen wurden, werden heute Störfälle im IoT mit Szenarien assoziiert, die komplette Industrieanlagen oder Kraftwerke lahmlegen. Der Bedarf an Netzsicherheit steigt parallel zum Ausbau und zur Nutzung des IoT.
Der Begriff „Internet of Things" wurde vorausschauend schon im Jahr 1999 geprägt. Die Zeit des explosiven Wachstums ist heute gekommen, denn inzwischen konnten die nötigen Voraussetzungen bei den Prozessoren und Netzwerken geschaffen werden. Ganz nach dem Mooreschen Gesetz sanken die Preise der Hardware drastisch, und der allgegenwärtige, kostengünstige Zugang zum Internet wurde selbstverständlich. Hinzu kommen die anhaltende Miniaturisierung der Komponenten sowie die fortschreitende Steigerung der Datentransport- und Verarbeitungsgeschwindigkeiten.

## Verteilte Verantwortung

Es wird höchste Zeit, das Sicherheitsniveau unter der Verantwortung aller Beteiligten nachhaltig anzuheben. Das Internet agiert ohne zentrale Verwaltung, es stützt sich auf das *Multistakeholder Model*, ein Rahmen, der jedem Beteiligten ein Mitwirkungsrecht einräumt. Die meisten Beteiligten lassen sich vier Hauptgruppen zuordnen: Die Regierungen, die Industrie, die Gemeinschaft auf technischer Ebene sowie die bürgerliche Gesellschaft. Zur letztgenannten Gruppe gehören auch jene Nutzer, die das Internet nur als Informationssystem oder für den Datentransport einsetzen. Grundsätzlich ist jeder, der sich an das Internet anbindet, daran beteiligt.
Die Steigerung der Sicherheit im Internet bedarf eines Wechselspiels zwischen allen Beteiligten. Erinnert sei an unzureichend gesicherte, IoT-fähige Geräte, die erst kürzlich in den Schlagzeilen waren: Smart-TVs, die unbemerkt das Sehverhalten des Zuschauers protokollieren oder sogar Gespräche aus dem Wohnzimmer weiterleiten, Kameras, die Aufnahmen ungesichert in das Internet streamen oder Spielzeugpuppen, die das Kinderzimmer ausspionieren. Das sind Beispiele für eindeutige Fälle, bei denen sich der Hersteller zu wenig um die Sicherheit gekümmert hat. Die Gründe liegen auf der Hand: *Time-to-Market* bedeutet, dass der Erste auf dem Markt die höchsten Umsätze für sich verbuchen kann. Folglich wird unter enormem Zeitdruck entwickelt und produziert, Aspekte der Sicherheit treten weit in den Hintergrund. Die Sicherheit im Internet ist mit dem Nimbus behaftet, Geld zu kosten, ohne Geld einzubringen. Der Käufer ist meist nicht bereit, einen höheren Preis für ein Produkt zu zahlen, das eine höhere Sicherheit bietet. Hinzu kommt, dass die IoT-Einsteiger unter den Herstellern konventioneller Produkte kaum mit Sicherheitsaspekten vertraut sind (zum Beispiel die Hersteller von Waschmaschinen oder Spielzeugpuppen). Die Gefahr, dass sich Fehler aus den Anfangstagen des Internets wiederholen, ist nicht von der Hand zu weisen. Dazu zählt das Ausliefern von Geräten wie Routern oder Telefonanlagen mit Standard-Passwörtern, ohne dem Käufer dringend nahe zu legen, das Passwort individuell zu ändern.

## Security by Design

Eine Antwort auf diese Herausforderung lautet: *Security by Design*. Die Hersteller IoT-fähiger Geräte müssen die Sicherheit als integralen Bestandteil in ihre Entwicklungen einbeziehen, anstatt nachträglich zu patchen. Weltweit arbeiten Entwickler an offenen Standards, die diesen Ansatz verfolgen. Auf der anderen Seite tragen auch die Nutzer ein Stück Verantwortung: Sie müssen ihre Möglichst-Billig-Mentalität revidieren und Geräte kaufen, die hohen Sicherheitsstandards genügen. Parallel dazu ist notwendig, dass der durchschnittliche Nutzer seine digitale Kompetenz erweitert. Elementare Grundkenntnisse in der Daten- und Informationstechnik genügen heute längst nicht mehr. Ein vertieftes Verständnis der Zusammenhänge ist notwendig, um Gefahren im Vorfeld zu erkennen und rechtzeitig Gegenmaßnahmen zu treffen.
Eine Aufgabe öffentlicher Bildungseinrichtungen ist das Vermit-



*Bild: Smh232. Bildquelle: Wikimedia.*

teln von Grundwissen und die Weiterbildung auf dem Gebiet der Informationstechnik. Verbraucherschutz-Organisationen können dem Käufer Entscheidungshilfen geben, indem sie die Sicherheit von IoT-Geräten ähnlich der Energie-Effizienz von Haushaltsgeräten bewerten und mit Labels kennzeichnen.

**Permissionless Innovation**

Der Gesetzgeber ist andererseits in der Pflicht, das IoT nicht durch ein Unmaß an Vorschriften und Regelwerken zu ersticken. Der Grundgedanke der Freiheit und Offenheit im Internet darf von keiner Seite ausgehöhlt werden. Bisher steht das Anbieten eines Dienstes im Internet jedem Interessenten offen. Gerade die *Permissionless Innovation*, also die Innovation ohne die Notwendigkeit einer hoheitlichen Absegnung, hat dazu beigetragen, dass sich das Internet rasant entwickeln konnte. Als Larry Page und Sergey Brin begannen, ihre Idee einer „Suchmaschine Google" in die Tat umzusetzen, bedurften sie keiner behördlichen Genehmigung. Auch das Realisieren eines individuellen Bereichs innerhalb des Internet steht jedem Interessenten frei. Für den Aufbau von Heim-Netzwerken und das Anbinden an das Internet ist weder eine Genehmigung noch eine Gerätezulassung nötig. Ganz anders als früher beim Telefonnetz, einem in sich geschlossenen System...

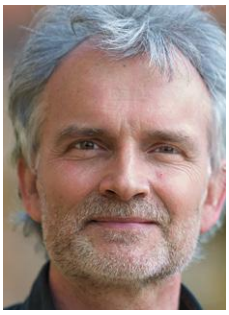
Die Sicherheit in der Informationstechnik muss deutlich angehoben werden, gleichzeitig darf der Charakter der Offenheit nicht leiden. Dass dies keine leichte Aufgabe ist, liegt in der Natur der Sache. Der Nutzer ist gefordert, die technischen Hintergründe besser zu verstehen, auch wenn er an dieser Materie vielleicht nur wenig Interesse hat. Die Hersteller müssen in die Sicherheit ihrer Produkte investieren, auch wenn sich dies nicht sofort in barer Münze auszahlt. Verantwortung für die Sicherheit müssen alle Beteiligten übernehmen, anderenfalls ist das Internet of Things potentiellen Cyberangriffen ohnmächtig ausgesetzt. Düstere Szenarien gehen sogar so weit, dass die globalen Infrastrukturen infolge von Sorglosigkeit im IoT vollständig zusammenbrechen. ◀

(160473)gd

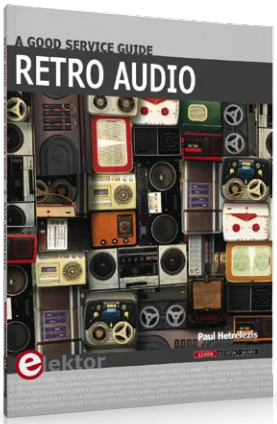
# Willkommen in Ihrem E-SHOP

ELEKTOR EMPFIEHLT

## Retro Audio



Mitten unter den vielen Veröffentlichungen von Elektor zu Themen wie Mikrocontroller, Software, Programmierung und Embedded-Boards erscheint da plötzlich ein Buch, das sich an Elektroniker richtet, die nicht ganz so scharf auf die allerneueste Elektronik sind. Paul Hetrelezis versorgt uns mit seinem Buch „Retro Audio – a Good Service Guide" mit dem nötigen Wissen und führt so an die wundervolle Welt, der Instandsetzung von Audio-Equipment aus der analogen Welt. Gemeint sind Tuner, Plattenspieler, Kassettenrekorder und Verstärker mit den Buchstaben, die vor der Klasse D kommen und mit diskreten Bauteilen wie Transistoren arbeiten, an denen man sich zur Not auch noch die Finger verbrennen kann.




**Jan Buiting** (Elektor-Redaktion)

www.elektor.de/retro-audio

## Elektor-Bestseller

1. Andonstar USB-Mikroskop A1
   www.elektor.de/andonstar-a1

2. ESP8266-WLAN-Funkmodul
   www.elektor.de/150445-91
3. Das MQTT-Praxisbuch
   www.elektor.de/das-mqtt-praxisbuch
4. StromPi 2
   www.elektor.de/strompi-2
5. Serielle Schnittstellen
   www.elektor.de/serielle-schnittstellen
6. Sensor-Kit X40
   www.elektor.de/sensor-kit-x40
7. Raspberry Pi 3
   www.elektor.de/rpi-3

## Das MQTT-Praxisbuch

MQTT ist ein leichtgewichtiges, ereignis- und nachrichtenorientiertes Protokoll zur effizienten und asynchronen Kommunikation zwischen Geräten auch über limitierte Netzwerke. Das bereits 1999 von IBM entwickelte Protokoll eignet sich heute in besonderer Weise für IoT-Anwendungen. Das MQTT-Praxisbuch führt Schritt für Schritt in die praktische Arbeit mit diesem ressourcensparenden Protokoll ein.

Mitgliederpreis: 34,80 €

www.elektor.de/das-mqtt-praxisbuch

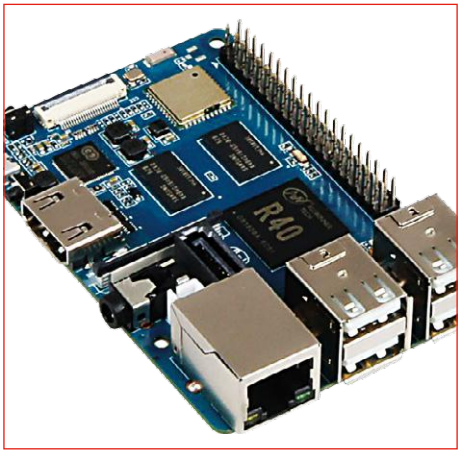
## Das ESP8266-Praxisbuch

Der ESP8266 ist ein programmierbares WLAN-Funkmodul mit zahlreichen Schnittstellen wie UART, I²C und SPI. Das Board ist sehr preiswert und bereits für unter 3 Euro verfügbar. Die UART-Schnittstelle sorgt dabei für eine einfache Integration in Mikrocontrollerprojekte. Das ESP8266-Modul kann hervorragend mit dem Arduino zusammenarbeiten und ermöglicht ihm über die serielle Schnittstelle den Zugang zum Netzwerk und Internet.

Mitgliederpreis: 39,80 €

www.elektor.de/esp8266-buch

## Banana Pi M2 Berry



Der Banana Pi M2 Berry ist baugleich mit dem Raspberry Pi 3 und voll kompatibel zum gesamten Raspberry Pi-Zubehör. Darüber hinaus bietet er eine Vielzahl an technischen Highlights. Er verfügt über eine GB-LAN-Schnittstelle, SATA-Anschluss und eine leistungsstarke ARM Cortex-A7 Quad-Core CPU (1200 MHZ). Er unterstützt folgende Betriebssysteme: Android, Debian Linux, Ubuntu Linux und Raspbian.

Mitgliederpreis: 38,66 €

www.elektor.de/banana-pi-m2-berry

# Arduino

**Überarbeitete Neuauflage!**

## Schaltungsprojekte für Profis

Mit diesem Buch wird der Leser umfassend in die Hardware und Software der Arduino-Plattform eingeführt. Einfache, leicht verständliche Projekte am Anfang des Buches führen Schritt für Schritt in die Open-Source-Plattform ein. Die Projekte werden dann zunehmend komplexer, um dem Leser möglichst viele konkrete Lösungsmöglichkeiten aufzuzeigen, die mit dem Arduino-Mikrocontroller auch für angehende Profis zur Verfügung stehen.

Dabei wird neben den erforderlichen theoretischen Grundlagen stets größter Wert auf eine praxisorientierte Ausrichtung gelegt. So werden wichtige Techniken wie AD-Wandlung, Timer oder Interrupts stets in Praxisprojekte eingebettet. Es entstehen Lauflichteffekte, ein Aufwachlicht, voll funktionsfähige Voltmeter, präzise Digitalthermometer, Uhren in allen Variationen, Reaktionszeitmesser oder mausgesteuerte Roboterarme. Und ganz nebenbei hat der Leser die Basics der zugehörigen Controller-Technik verstanden und im wahrsten Sinne des Wortes begriffen.

**Mitgliederpreis: 33,00 € (frei Haus)**
**www.elektor.de/arduino-schaltungsprojekte**

## Serielle Schnittstellen

Die serielle Schnittstelle dient dem Datenaustausch zwischen Mikrocontrollern und Peripheriegeräten. Das Wissen um diese seriellen Schnittstellen erweitert die Einsatzmöglichkeiten von Mikrocontrollern und Peripheriegeräten ungemein. In diesem Buch werden die Grundlagen der Kommunikation für die drei seriellen Schnittstellen I²C, SPI und 1-Wire vorgestellt.

**Mitgliederpreis: 39,80 €**

**www.elektor.de/serielle-schnittstellen**

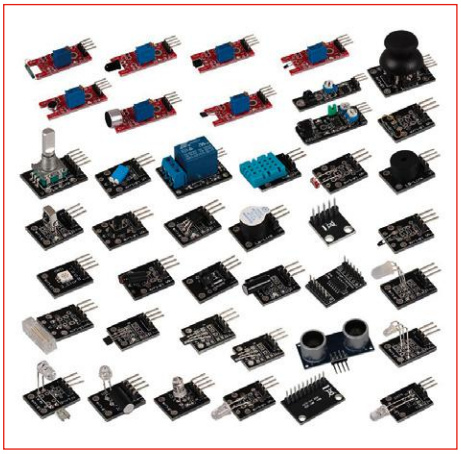
## AVR-Programmierung für Quereinsteiger

Atmel AVR ist eine 8-bit-Mikrocontroller-Familie des Herstellers Atmel. Diese Controller sind wegen ihres einfachen Aufbaus, ihrer leichten Programmierbarkeit, den kostenlosen Entwicklungswerkzeugen und der Verfügbarkeit in DIP-Gehäuseformen auch bei Makern äußerst beliebt. Darüber hinaus sind diese Controller bereits ab zwei Euro erhältlich. Im Arduino Uno-Board wird der ATmega328 verwendet.

**Mitgliederpreis: 34,80 €**

**www.elektor.de/avr-prog-fuer-quereinsteiger**

## Sensor-Kit X40



Dieses hochwertige Sensor-Kit wurde speziell für die gängigsten Open-Source-Plattformen entwickelt. Es ist kompatibel zu Einplatinencomputern (Raspberry Pi, Banana Pi, CubieBoard, CubieTruck, Beaglebone, pcDuino) und Mikrocontrollern (Arduino, ATmega, AVR, PIC, STM32 usw.). Es beinhaltet insgesamt 40 verschiedene Sensoren. Sie können die Sensoren entweder fest verlöten oder auf ein Breadboard stecken.

**Mitgliederpreis: 71,96 €**

**www.elektor.de/sensor-kit-x40**

# Hexadoku

## Sudoku für Elektroniker

Wie in jeder Ausgabe finden Sie auch in diesem Heft unser ganz spezielles Sudoku. PC, Oszilloskop und Lötkolben können sich erholen, während Ihre kleinen grauen Zellen auf Hochtouren arbeiten. Wenn Sie alle Hex-Ziffern in den grauen Kästchen herausgefunden haben, sollten Sie uns diese gleich zumailen – denn hier warten fünf Elektor-Gutscheine!

Die Regeln dieses Rätsels sind ganz einfach zu verstehen: Bei einem Hexadoku werden die Hexadezimalzahlen 0 bis F verwendet, was für Elektroniker und Programmierer ja durchaus passend ist. Füllen Sie das Diagramm mit seinen 16 x 16 Kästchen so aus, dass alle Hexadezimalzahlen von 0 bis F (also 0 bis 9 und A bis F) in jeder Reihe, jeder Spalte und in jedem Fach mit 4 x 4 Kästchen (markiert durch die dickeren schwarzen Linien) **genau einmal** vorkommen. Einige Zahlen sind bereits eingetragen, was die Ausgangssituation des Rätsels bestimmt. Wer das Rätsel löst – sprich die Zahlen in den grauen Kästchen herausfindet – kann einen von fünf Gutscheinen im Wert von 50 Euro gewinnen!

### Einsenden

Schicken Sie die Lösung (die Zahlen in den grauen Kästchen) per E-Mail, Fax oder Post an:

**Elektor Redaktion**
**Kackertstr. 10**
**52072 Aachen**

**Fax: 0241 / 955 09-013**
**E-Mail: hexadoku@elektor.de**

Als Betreff bitte nur die Ziffern der Lösung angeben!

**Einsendeschluss ist der 21. September 2017.**

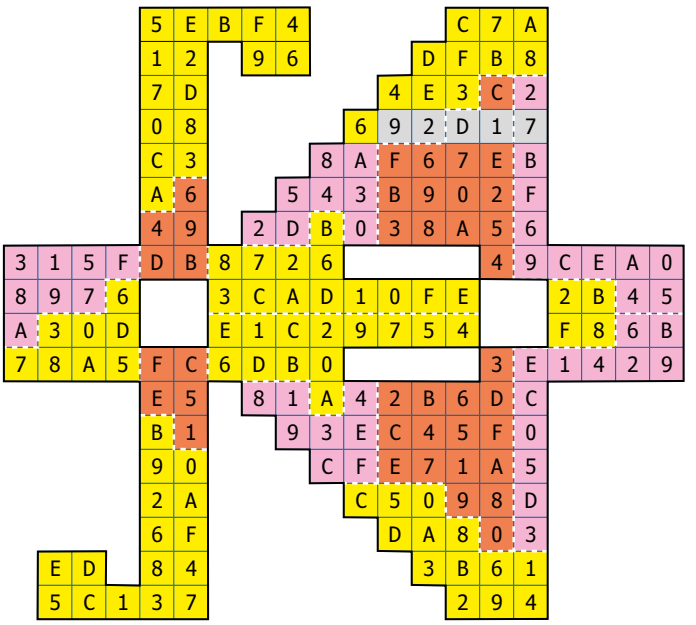
### Die Gewinner des Hexazeners aus der Juli/August-Ausgabe stehen fest!

Die richtige Lösung ist: **92D17.**

Einen Elektor-Wertgutschein über je 50 € haben gewonnen: Gilbert Luyckx, Jean-Claude Carré, Harrie Rouwhorst, Per Troelsen und Chris Cliquet.

Herzlichen Glückwunsch!

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | E | | | C | | | 9 | | | | | | 6 | | F |
| | | 6 | | | | | | 9 | C | 4 | | A | | 0 | 1 |
| | F | | | 0 | | | | | 6 | D | 8 | 9 | 2 | B | |
| | | | 1 | 3 | | | | 7 | E | | | | | | 8 |
| B | 0 | A | | | | 2 | 1 | | | 3 | F | 6 | | | D |
| D | | | | | | | 0 | 6 | | | 1 | B | 5 | | |
| F | | | | | | 4 | D | E | 8 | A | C | | | | 0 |
| | 4 | 3 | | 7 | | 8 | | | | | | | 1 | | |
| | | E | | | | | | | B | | 3 | | 4 | 1 | |
| 2 | | | | B | 0 | 6 | E | F | A | | | | | | 3 |
| | | 8 | B | 1 | | | 4 | 2 | | | | | | | 6 |
| 6 | | | 4 | F | 2 | | | 5 | 9 | | | | 7 | A | B |
| 1 | | | | | | 0 | 5 | | | | 9 | 2 | | | |
| | 9 | 7 | C | E | 3 | 1 | | | | | 0 | | | F | |
| 5 | 2 | | A | | 9 | F | 8 | | | | | | 0 | | |
| 4 | | D | | | | | | 1 | | | 5 | | | 3 | |



*Der Rechtsweg ist ausgeschlossen. Mitarbeiter der in der Unternehmensgruppe Elektor International Media B.V. zusammengeschlossenen Verlage und deren Angehörige sind von der Teilnahme ausgeschlossen.*

# Join the elektor Community

**Lesen Sie jetzt die europaweit umfassendste deutschsprachige Fachzeitschrift für Elektronik und Mikrocontroller in der ultimativen GOLD-Mitgliedschaft und profitieren Sie von allen Vorteilen!**

## Ihre Vorteile als GOLD-Mitglied:

- 6x Elektor-Doppelausgabe (132 Seiten!) pünktlich und zuverlässig frei Haus
- Jedes Heft gleichzeitig als PDF unterwegs lesen
- 6x Elektor Business gedruckt und digital
- Zugriff auf das gesamte Elektor-Archiv ab 1990
- Zugang zu Gerber-Dateien von über 1.000 Labs-Projekten
- Jahrgangs-DVD (Wert: 27,50 €) ohne Zuzahlung
- 10% Online-Rabatt auf alle Shop-Produkte

elektor
GOLD Mitgliedschaft

GREEN Mitgliedschaft

**UMWELTSCHONEND – GÜNSTIG – GREEN**

**Möchten Sie Elektor lieber im elektronischen Format beziehen? Dann ist die GREEN-Mitgliedschaft ideal für Sie!**

**Die GREEN-Mitgliedschaft bietet (abgesehen von den Printausgaben) alle Leistungen und Vorteile der GOLD-Mitgliedschaft.**

**Jetzt Mitglied werden unter**
**www.elektor.de/mitglied**

# reichelt elektronik

Technik verbindet.

**BESTELLT | BELIEFERT | BEGEISTERT**

- ✓ Mehr als 80.000 Produkte
- ✓ Top-Preis-Leistungsverhältnis
- ✓ Hohe Verfügbarkeit und 24-h-Lieferservice
- ✓ Starke Marken und echte Qualität

reichelt.de | @reichelt_el | google.com/+reichelt

# BUILD, REPAIR & SAVE
## LIKE NEVER BEFORE — RND

**RND steht für gute Qualität, günstige Preise und Zuverlässigkeit.**

Zugeschnitten auf den professionellen Einsatz entsprechen die Produkte genau den Anforderungen im Beschaffungsmanagement sowie den Bedürfnissen technisch hoch versierter Anwender und Ingenieuren der Elektrotechnik.

Besuchen Sie regelmäßig den reichelt Online-Shop und entdecken Sie die neuesten Produkte von

RND connect, RND cable, RND components & RND lab.

RND

### Lötbare Schraubklemme
RND connect

2-pol, RM 5,08 mm, 90°

FLACHE BAUFORM
cUL-ZULASSUNG

- Stift/Klemme - verzinntes Messing
- Schraube/Drehmoment - M2,5 Stahl, vernickelt

Bestell-Nr.: RND 205-00232
**0,24**

**>400** Schraubklemmen, Steckverbinder & Stiftleisten
JETZT ENTDECKEN ▶ http://rch.lt/Fp

### PVC-Litze
RND cable

0,75 mm², 100 m, rot

H05V-K

- flexibler Kupferleiter Klasse 5
- fester Temperaturbereich -30 °C bis +70 °C
- Prüfspannung: 300/500V

Bestell-Nr.: RND 475-00101
**20,40**

**>300** Gummi- & Kunststoff-Schlauchleitungen & Erd- & Einzeladern
JETZT ENTDECKEN ▶ http://rch.lt/Fo

### Sicherungshalter
RND components

5 x 20 mm, geschlossen

- 100 MOhm Isolationswiderstand
- Raster 22,6 mm
- 1,5 mm Bohrungsdurchmesser in Leiterplatte

Bestell-Nr.: RND 170-00004
**0,24**

**>2100** Widerstände, Sicherungen & Gehäuse
JETZT ENTDECKEN ▶ http://rch.lt/Fq

### Präzisionspinzetten-Set
RND lab

5-teilig

- aus rostfreiem Stahl, Typ SA
- nicht magnetisierbar
- gute Korrosionsbeständigkeit gegenüber den meisten Chemikalien, Salzen und Säuren

Bestell-Nr.: RND 550-00011
**50,45**

**NEU** schon **60** Endoskope, Mikroskope, Pinzetten, Boxen & Koffer
JETZT ENTDECKEN ▶ http://rch.lt/Fr

Tagespreise · Preisstand: 24. 7. 2017, Preise in € inkl. gesetzl. MwSt., zzgl. Versandkosten, reichelt elektronik, Elektronikring 1, 26452 Sande (D)

**JETZT NEWSLETTER ABONNIEREN & GEWINNEN!**
Wir verlosen monatlich unter allen neuen Newsletter-Abonnenten ein technisches Highlight!
JETZT MITMACHEN ▶ http://rch.lt/v3

**www.reichelt.de**
BESTELLHOTLINE: +49 (0)4422 955-333